Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 3159-0040
Vol. 2 Issue 11, November - 2015

Performance Analysis of Hummingbird Cryptographic Algorithm using FPGA

Miss. Pranali Umap
Electronics And Telecommunication,
SIPNA C.O.E.T/ SGBAU University,
Amravati (Maharashtra State), India.
umap.pranali@gmail.com

Dr. A.S.Joshi
Professor, Electronics And Telecommunication,
SIPNA C.O.E.T/ SGBAU University
Amravati (Maharashtra State), India

***Abstract*—Hummingbird is a novel ultra-lightweight Cryptographic Algorithm aiming at resource-constrained devices. It has a hybrid structure of block cipher and stream cipher and was developed with both lightweight software and lightweight hardware implementations for constrained devices in mind. Moreover, Hummingbird has been shown to be resistant to the most common attacks to block ciphers and stream ciphers including birthday attack, differential and linear cryptanalysis, structure attacks, algebraic attacks, cube attacks, etc. This paper provides the issues in Hummingbird Cryptographic Algorithm for efficiency, logic elements and throughput. This also introduces the concept of pipelining which improves the speed of execution. In this project, we have optimizes some blocks to get higher throughput and higher efficiency. This modification result into faster encryption and decryption of the message. Here the VHDL code is written for encryption and decryption of hummingbird cryptosystem. It is compiled and simulated by using ModelSim SE 6.3f, synthesized by using Altera Quartus II 9.1 sp2.**

Keywords— Cryptography, Block cipher, Encryption, Decryption

I. INTRODUCTION

Cryptography is the art or science of hiding the information content using a key. Cryptographic algorithms are ubiquitous in modern communication systems where they have a central role in ensuring information security. Low-cost smart devices like RFID tags and smart cards are rapidly becoming important in our daily life. Well known applications include electronic passports, contactless payments, product tracking, access control and supply chain management just to name a few. But the small programmable chips that passively respond to every reader have raised concerns among researchers about privacy and security breaches. A considerable body of research has been focused on providing RFID tags with cryptographic functionality. LWC does not determine strict criteria for classifying cryptographic algorithms as lightweight, but the common features of lightweight algorithms are extremely low requirements to essential resources of target devices. Hummingbird is a new lightweight cryptographic algorithm targeted for resource constrained devices since the traditional algorithm like AES,SEA were not suitable for implementation in low resource devices like wireless sensor nodes, smart card, RFID tags.

Hummingbird is a novel ultra-lightweight cryptographic algorithm aiming at resource-constrained devices. It has a hybrid structure of block cipher and stream cipher and developed with both lightweight software and lightweight hardware implementations for constrained devices in mind. Moreover, Hummingbird has been shown to be resistant to the most common attacks to block ciphers and stream ciphers including birthday attack, differential and linear cryptanalysis, structure attacks, algebraic attacks, cube attacks, etc.

Different from existing (ultra-)lightweight cryptographic primitives which are either block ciphers or stream ciphers, Hummingbird is an elegant combination of the above two cipher structures with a 16-bit block size, 256-bit key size, and 80-bit internal state. The design of the Hummingbird encryption scheme is motivated by the well-known Enigma machine and takes into account both security and efficiency simultaneously. The encryption /decryption process of the Hummingbird can be viewed as the continuous running of a rotor machine, where four small block ciphers act as four virtual rotors which perform permutations on 16-bit words. Moreover, extremely simple arithmetic and logic operations are extensively employed in the Hummingbird, which make it well-suited for resource-constrained environments.

The design of the Hummingbird encryption scheme is motivated by the well-known Enigma machine and takes into account both security and efficiency simultaneously. The encryption and decryption process of the Hummingbird can be viewed as the continuous running of a rotor machine, where four small block ciphers act as four virtual rotors which perform permutations on 16-bit words. The salient characteristics of the Hummingbird lies in implementing extraordinarily large virtual rotors with custom block ciphers and using successively changing internal states to step each virtual rotor in various and unpredictable ways.

Proposed Hummingbird cryptographic algorithm is very efficient algorithm which uses the concept of pipelining to enhance the throughput and efficiency of the encryption and decryption process. In this project, at some instances we have optimizes some blocks by using simple XOR operation instead of traditional modulo addition which causes the reduction in logic elements which results to faster encryption and decryption of the message.

II. RELATED WORK

Xinxin Fan and Guang Gong, Ken Lauffenburger and Troy Hicks [1] provides the efficient hardware implementations of a standalone Hummingbird component in field-programmable gate array (FPGA) devices. The implementation includes encryption only core and an encryption/decryption core on the low-cost Xilinx FPGA series **Spartan-3** The experimental results highlight that in the context of low-cost FPGA implementation Hummingbird has favorable efficiency and low area requirements. The experimental results provides the area requirement (in slices),the maximum work frequency and throughput in field-programmable gate array.

Reena Bhatia [2] presented the enhanced hardware implementation of the Hummingbird cryptographic algorithm for low-cost Spartan-3 FPGA family. It gives the efficient FPGA implementations of a standalone Hummingbird component. The implementations includes an encryption only core and an encryption/decryption core on the low-cost Xilinx FPGA series Spartan-3 and comparisons of the results with other reported (ultra-) lightweight block cipher implementations on the same series. The coprocessor approach is enabled due to the fact that FPGAs have dedicated memory blocks. The datapath of the Hummingbird coprocessor is implemented in four stages and the instruction count is reduced via pipelining technique.

Nikita Arora and Yogita Gigras [3] provides a low power and high speed lightweight cryptographic Hummingbird algorithm for hardware environment. The performance of the approach used is determined on XILINX platform using Verilog as hardware description language. Hummingbird cryptographic algorithm for low power and high operating speed is performed using Virtex5 family of XILINX ISE suite. This emphasizes the hardware implementation of the Hummingbird algorithm, so the FPGA(Field Programmable Gate Arrays) is the hardware platform selected depending on the application needs and constraints. The design can be implemented on every electronic system which is the part of mobile adhoc network to prevent the security breach.

Revini S. Shende and Mrs. Anagha Y. Deshpande [4] presented the algorithms for the encryption as well as decryption process and shows some simulation results performed on Xilinx**.** It includes details about lightweight cryptography and its types and discusses the implementation of ultra lightweight cryptographic algorithm Hummingbird. The security and performance factor is very precisely achieved by the algorithm due to its prominent internal structure. The efficient FPGA implementation of Hummingbird is possible using the given software algorithms so that it can achieve larger throughput with smaller area requirement. Also, Hummingbird can be used in high-security required devices as it is resistant to most cryptographic attacks**.**

M. Rabbani and R. Ramprakash [5] provides the design of Hummingbird algorithm for advanced crypto systems which includes round based architecture of 16-bit block cipher. It implements an encryption and decryption core on the low cost Xilinx FPGA series Spartan-3.This gives the technique to reduce number of clock cycles to encrypt and decrypt the message. This reduction in number of clock cycles allows faster encryption and decryption of the message. As compared to other lightweight FPGA implementation of block cipher AES, Hummingbird can encrypt and decrypt in less number of clock cycles.

Sergey Panasenko and Sergey Smagin [6] presented the underlying principles and approaches used in lightweight cryptography. In this paper they propose generalized approaches to lightweight algorithms design. Also, they highlight some constraints and recommendations for implementation of lightweight algorithms. Finally, they anticipate several trends in lightweight cryptography. In this paper they consider lightweight block ciphers and propose generalized approaches to lightweight algorithms design. The paper highlight some constraints and recommendations for implementation of lightweight algorithms. Also, it describe compromises which should be reached by designers of lightweight cryptographic primitives. Finally, they anticipate several trends in lightweight cryptography.

Jacob John [7] presented the difference between the Humming Bird-2 and PRESENT cryptographic algorithm. Humming Bird-2 and PRESENT are two recently proposed Light weight cryptographic algorithms specifically made for implementation in resource constrained devices like wireless sensor nodes, smart cards and RFID systems. Performance analyses of these two efficient algorithms are done in this paper. It is found from security analysis that both the algorithms provide adequate security. The hardware implementation of PRESENT requires comparatively less area. PRESENT is more efficient in FPGA implementation. But the power consumption of Humming bird-2 is comparatively less in hardware implementation. The throughput of Humming bird-2 is higher than other algorithms. So it is found from the analysis that Humming bird-2 is more suitable as

cryptographic algorithm for resource constrained devices.

Daniel Engels, Markku-Juhani O. Saarinen,Peter Schweitzer, and Eric M. Smith [8] providesHummingbird-2 is an encryption algorithm with a 128-bit secret key and a 64-bit initialization vector. Hummingbird-2 optionally produces an authentication tag for each message processed. Like it's predecessor Hummingbird-1, Hummingbird-2 has been targeted for low-end microcontrollers and for hardware implementation in lightweight devices such as RFID tags and wireless sensors. Hummingbird-2, a lightweight authenticated encryption algorithm and believed to be resistant to all standard attacks to block ciphers and stream ciphers such as differential and linear cryptanalysis, structure attacks and various algebraic attacks.

Ashwani Sengar, Prince Nagar and Mayank Sharma[9] presented Hummingbird is a new ultra-lightweight crypto-graphic algorithm targeted for resource-constrained devices like RFID tags, smart cards, and wireless sensor nodes. This paper gives design of algorithm using Dsp module, it also describe efficient hardware implementations of a Hummingbird component in field-programmable gate array (FPGA) devices. It include the implementation of an encryption/decryption core on the low-cost Xilinx FPGA series Vertex-5 and compared the results with other reported lightweight block cipher implementations. The experimental results highlight that in the context of low-cost FPGA implementation Hummingbird has favorable efficiency and low area requirements.

III. Proposed Work

Hummingbird algorithm:

Hummingbird is neither a block cipher nor a stream cipher, a rotor machine equipped with novel rotor design of Hummingbird is based on an elegant combination of a block cipher and stream cipher with 16 bit key size, and 80-bit internal state. The size of the key and the internal state of Hummingbird provides which is adequate for many embedded Applications. The steps performed are:

A. Initialization process:

Figure 1. Shows the overall structure of Hummingbird Initialisation Algorithm. When using Hummingbird in practice, four 16-bit random nonces NONCE*i* are first chosen to initialize the four internal state registers *RSi* (*i* = 1*;* 2*;* 3*;* 4) respectively followed by four consecutive encryptions on the message *RS*1 _ *RS*3 by Hummingbird running in initialization mode . The final 16- bit ciphertext *TV* is used to initialize the LFSR. Moreover, the 13th bit of the LFSR is always set to prevent a zero register. The LFSR is also stepped once before it is used to update the internal state register *RS*3.

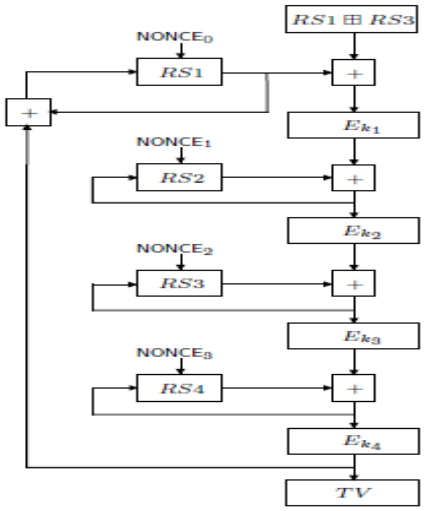


Fig. 1: Initialization process

Algorithm1:Hummingbird Initialization

Input: Four 16 bit random nonce NONCE (i=1,2,3,4)

Output: Initialized four rotors RSI4(i=1,2,3,4) and LFSR

B. *Encryption Process:*

The overall structure of the Hummingbird encryption algorithm is depicted in Figure 2. After a system initialization process, a 16-bit plaintext block *PTi* is encrypted by first executing a modulo 216 addition of PTi and the content of the first internal state register *RS*1. The result of the addition is then encrypted by the first block cipher *Ek*1. This procedure is repeated in a similar manner for another three times and the output of *Ek*4 is the corresponding ciphertext CTi. Furthermore, the states of the four internal state registers will also be updated in an unpredictable way based on their current states, the outputs of the first three block ciphers, and the state of the LFSR

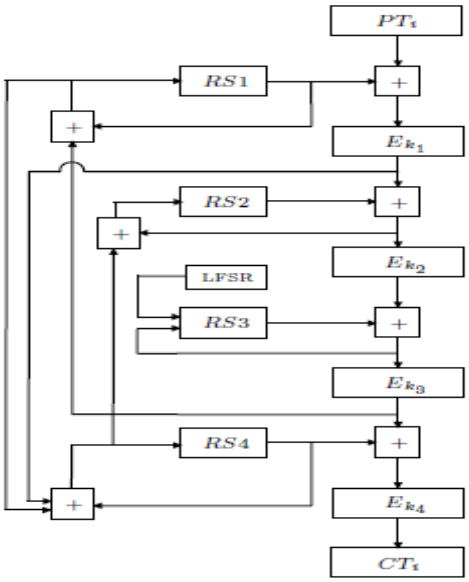


Fig. 2: Encryption process

Algorithm 2: Hummingbird Encryption
Input: A 16 bit plaintext PTi and four rotors RSIt (i=1,2,3,4)
Output: A ciphertext CTi

Encryption Process
$V12_t = E_{K1}(PT_i \oplus RS1_t)$
$V23_t = E_{K2'}(V12_t \oplus RS2_t)$
$V34_t = E_{K1'}(V23_t \oplus RS3_t)$
$CT'_t = E_{K2}(V34_t \oplus RS4_t)$
$CT_t = CT'_t \oplus RS1_t$

C. Decryption Process:

The overall structure of the Hummingbird decryption algorithm is illustrated in Figure below. The decryption process follows the similar pattern as the encryption and a detailed description is shown in the following Algorithm 3.

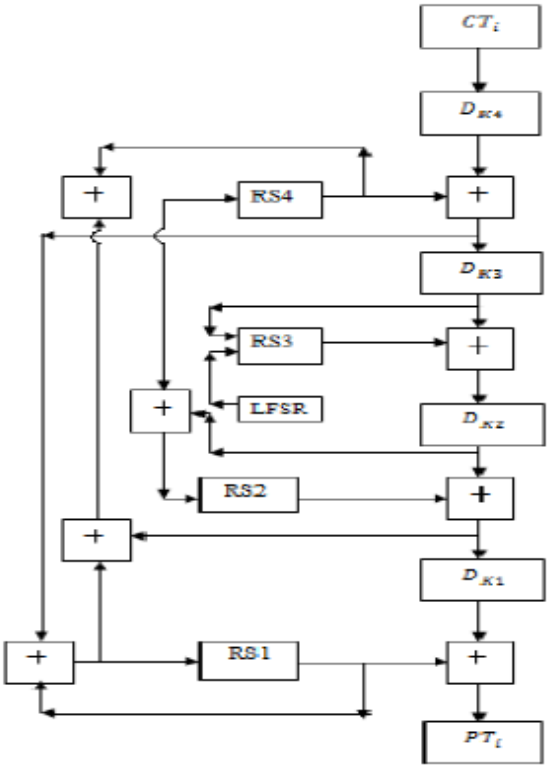


Fig:3 Decryption process

Algorithm 3: Hummingbird Decryption
Input: A 16 bit ciphertext CTi and four rotors RSIt (i=1,2,3,4)
Output: A 16 bit plaintext PTi

Decryption Process
$CT'_t = CTi \oplus RS1_t$
$V34_t = D_{K2}(CT'_t) \oplus RS4_t$
$V23_t = D_{K1'}(V34_t) \oplus RS3_t$
$V12_t = D_{K2'}(V23_t) \oplus RS2_t$
$PT_t = D_{K1}(V12_t) \oplus RS1_t$

D. 16-Bit Block Cipher:

Hummingbird employs four identical block ciphers *Eki* (*i* = 1*;* 2*;* 3*;* 4) in a consecutive manner, each of which is a typical substitutionpermutation (SP) network with 16-bitblock size and 64-bit key as shown in the following.The block cipher consists of four regular rounds and a final round. The 64-bit subkey *ki* is split into four 16-bit round keys keys *K*(*i*)5 and *K*(*i*)6 directly derived from the four round keys. While each regular round comprises of a key mixing step,a substitution layer, and a permutation layer,the final round only includes the key mixing and the S-box substitution steps. The key mixing step is implemented using a simple exclusive-OR operation,whereas the substitution layer is composed of four S-boxes with 4-bit inputs and 4-bit outputs.The selected four S-boxes, denoted by *Si*(*x*) : F42→F42*; i* = 1*;* 2*;* 3*;* 4, are Serpent that the 16-bit block cipher is resistant to linear and differential attacks as well as interpolation attack type S-boxes [1] with additional properties which can ensure.

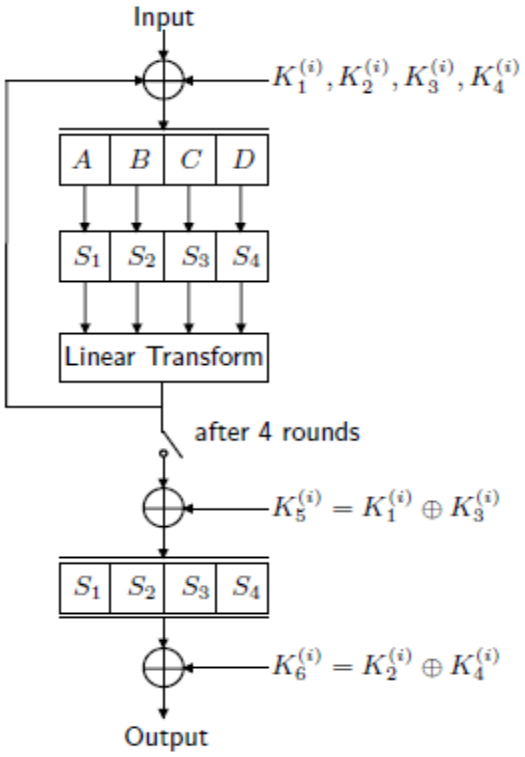
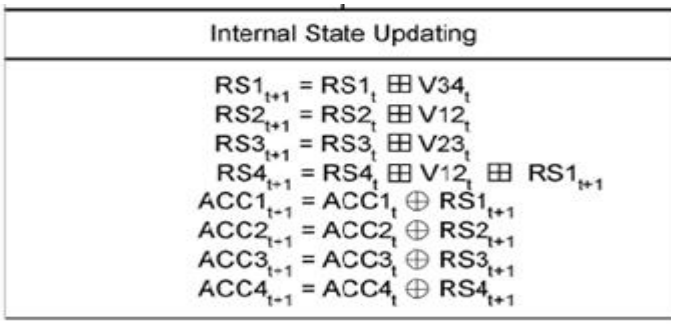


Fig.4: 16 bit Block cipher

Algorithm 4: 16 bit Block cipher

Input: A 16 bit data block m = (m0, m1, m2,……,m15) and a 64 bit subkey Ki (i= 1,2,3,4)
Output: A 16 bit data block m'=(m'0,m'1,……..m'15)



For the implementation of Hummingbird cryptographic algorithm the feasible and available platforms are FPGA, ModelSim, Xilinx ISE suite. FPGA configuration is specified using a hardware descriptive language. Verilog is the hardware description language used for designing as well as simultation purpose. The hardware design of hummingbird on FPGA is shown using hardware description language as verilog via After simulation and synthesis the next step is place and route which provides the hardware design for the proposed hummingbird cryptographic algorithm.

IV. Result & Discussion

A. *Simulation of Hummingbird Encryption*

The Hummingbird encryption is simulated on Modelsim simulator. The figure 5 shows the simulation result of encryption. The 16 bit plain text and four16 bit keys are the input and provides 16 bit cipher text as output.

I. The simulated result of hummingbird encryption:

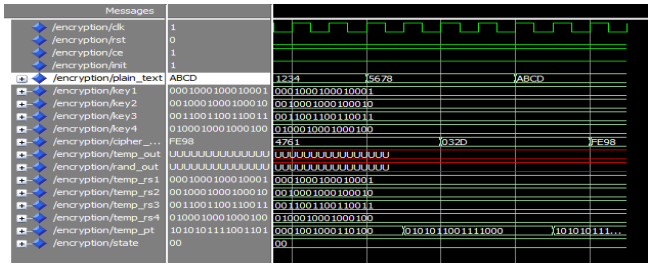


Fig5: Simulation Of Hummingbird Encryption.

B. *Simulation of Hummingbird Decryption*

The Hummingbird decryption is simulated on Modelsim simulator. The figure 6 shows the simulation result of decryption. The simulated result of hummingbird decryption:

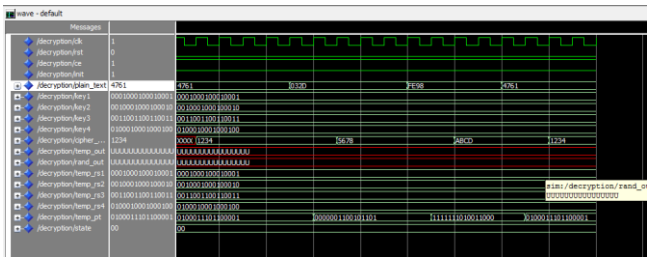


Fig 6: Simulation Of Hummingbird Decryption.

C. *Simulation of Hummingbird Cryptosystem*

The figure 7 shows the simulation result of cryptosystem. The Table I shows the inputs and output of the Cryptosystem.

Table I Hummingbird Cryptosystem Plane text, Cipher text and Recovered plain text

16 bit plain text	1010101111001101(ABCD)
16 bit block cipher	101001011110100(D2F4)
16 bit cipher text(encrypted)	1110000010100000(E0A0)
16 bit plain text(Recovered	1010101111001101(ABCD)

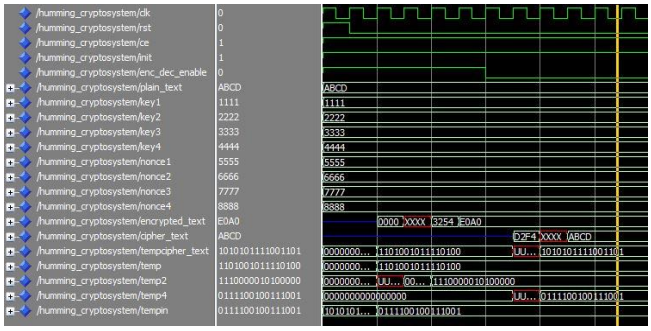


Fig 7: Simulation of Hummingbird Cryptosystem

D. *Analysis and Synthesis of Hummingbird Cryptosystem:*

A summary of our implementation results is presented in table II where the area requirement(in logic elements) is given. All experimental results were extracted using 9.1 build 350 SP 2 SJ on a EP2C20F484C7 quartus II

Table II: Device utilization summary using Altera Quartus II Cyclone II for Cryptosystem:

Resourses	Available	Used	Utilization
Total Logic Elements	18752	420	2%
Total Combinational Functions	18752	260	2%
Total Dedicated Registers	18752	240	1%
Total Pins	315	181	52%

V. APPLICATION

- Hummingbird Cryptographic Algorithm used for low-cost and low-power tags such as animal identification, point-of-sales, inventory management.
- It is used in low resource devices like wireless sensor nodes, smart cards and RFID tags.
- Hummingbird can be used in high-security required devices as it is resistant to most cryptographic attacks.

VI. CONCLUSION

The design of Proposed Hummingbird Cryptographic Algorithm is based on an elegant combination of a block cipher and stream cipher with 16- bit block size, 256-bit key size, and 80-bit internal state. The size of the key and the internal state of Hummingbird provides a security level which is adequate for many embedded Applications. In this project instruction count is reduced via pipelining technique. The proposed efficient hummingbird encryption/ decryption cores can encrypt or decrypt a 16 bit message block with four clock cycles. In this proposed work, we have optimizes some blocks by replacing traditional modulus operator by XOR operation, which causes the reduction in logic elements which results to faster encryption and decryption of the message. Compared to other lightweight block ciphers AES, SEA, proposed Hummingbird can achieve larger throughput and higher efficiency with the smaller area requirement. Consequently, proposed Hummingbird can be considered as an ideal cryptographic primitive for resource constrained environment. The proposed Hummingbird is possible using the given software algorithm so that it can achieve larger throughput and higher efficiency with smaller area requirement.

VII. FUTURE ASPECTS

Hummingbird Cryptographic algorithm have been designed as mutual authentication algorithm, which is a combination of both block cipher and stream cipher, is designed with a small block size and expected to meet stringent response time and power consumption. The prime focus is on the FPGA implementations of optimized novel hardware architectures and algorithms One could work on replacing four S-boxes used in Hummingbird Cryptographic algorithm by single S-box, which is repeated four times in the 16-bit block cipher. Furthermore study of optimization approaches for the implementations supporting separation of ultra-lightweight algorithms into a specific branch of lightweight cryptography and deepening the divergence between software and hardware oriented lightweight algorithms have tremendous scope for future work.

VIII. REFERENCES

[1]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.

[2]Reena Bhatia, "Study of Hummingbird Cryptographic Algorithm based on FPGA Implementation",2014 IJCSIT International Journal of Computer Science and Information Technologies,Vol.5(3),2014,4426-4430.

[3]Nikita Arora and Yogita Gigras, "FPGA Implementation of Low Power and High Speed Hummingbird Cryptographic Algorithm",International J Computer Applications(0975-8887) Volume 92-No.16,April 2014,42

[4]Revini S. Shende, Mrs. Anagha Y. Deshpande, "VLSI Design of Secure Cryptographic Algorithm",International Journal of Engineering Research and Applications(IJERA) ISSN:2248-9622 Vol. 3,Issue 2,March pp.742-746.

[5]M. Rabbani and R. Ramprakash, "Design of Hummingbird Algorithm for Advanced Crypto Systems",2014 IJEDR, Volume 2,Issue 1,ISSN:2321 9939,385-387.

[6]Sergey Panasenko and Sergey Smagin,"Lightweight Cry Underlying Principles and Approaches" International Journal of Computer Theory and Engineering, Vol. 3, No. 4, August 2011

[7] Jacob John Performance Analysis of New Light Weight Cryptographic Algorithms IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661, ISBN: 2278-8727 Volume 5, Issue 5 (Sep-Oct. 2012), PP 01-04

[8] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith, "Hummingbird: Ultra-Lightweight Cryptography for Resource- Constrained Devices", to appear in the proceedings of The 14th International Conference Cryptography and Data Security - FC 2010, 2010.

[9]Ashwani Sengar, Prince Nagar and Mayank Sharma Scholars Research Library Archives of Applied Science Research, 2013, 5 (3):270-277.