A Survey on Design of Hummingbird Cryptographic Algorithm

Pranali R. Umap **Dr. A.S. Joshi**

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 present the algorithms for the encryption as well as decryption process. It also discusses the concept of pipelining which results in reduction of number of clock cycles. This increases the throughput of the algorithm which gives fast encryption and decryption of the message.

Index Terms- Cryptography, Hummingbird Algorithm, Block Cipher, Attack, Data Security, Encryption, Decryption.

I. INTRODUCTION

Cryptography is the art or science of hiding the information content using a key. The techniques used for achieving this are Secret-Key Cryptography, Public-Key Cryptography and Elliptic Curve cryptography which are based on ASIC and are used in General purpose processors. A Low cost smart devices like RFID tags and smart cards are rapidly becoming pervasive in our daily life. Well known applications include electronic passports, contactless payments, product tracking just to name a few. But the small programmable chips that passively respond to every reader have raised concern among researchers about privacy and security. This emerging research area is usually referred to as lightweight cryptography[8].Lightweight cryptography is a branch of modern cryptography, which covers cryptographic algorithms indended for use in devices with low or extremely low resourses[9].

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.

Field Programmable Gate Arrays are programmable logic devices which have proven to be highly feasible implementation platforms for cryptographic algorithms because they provide both speed and programmability. The most important and the advanced one out of FPGA implementations is the Hummingbird algorithm. So the FPGA is the hardware platform selected depending on the application needs and constraints. FPGA configuration is specified using a hardware description language. Verilog is the hardware description language used for designing as well as simulation purposes. FPGAs comprises of logic blocks (flip flops, gates, memory elements)that are used to implement any logic functionality.

II. LITERATURE REVIEW

Hummingbird has been implemented across a wide range of different target platforms. Those implementations demonstrate that Hummingbird provides efficient and flexible software and hardware solutions for various embedded applications. There has been extensive previous work in the area of designing FPGA based system for Hummingbird cryptographic Algorithm.

Xinxin Fan and Guang Gong, Ken Lauffenburger and Troy Hicks [1] gives the efficient hardware implementations of a standalone Hummingbird component in field-programmable gate array (FPGA) devices The experimental results highlight that in the context of low-cost FPGA implementation Hummingbird has favorable efficiency and low area requirements. The proposed speed optimized Hummingbird encryption/ decryption cores can encrypt or decrypt a 16-bit message block with 4 clock cycles, after an initialization process of 20 clock cycles.

Reena Bhatia [2] proposed 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.

Nikita Arora and Yogita Gigras [3] proposed low power and high speed lightweight cryptographic Hummingbird algorithm for hardware environment. The implemented design consumes low power of 262.57 mW for 2.5 V with the operating speed or frequency of 152.905 MHz. The low power and High speed FPGA implementation is very precisely achieved by the proposed algorithm due to its prominent internal structure. Hence this high performance ultra-lightweight hybrid model will meet the power consumption requirements with constricted response time for diverse embedded applications and can be widely suitable for hardware environment.

Revini S. Shende and Mrs. Anagha Y. Deshpande [4] proposed the algorithms for the encryption as well as decryption process.

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] proposed the 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. hummingbird algorithm is fast in encrypting and decrypting the bits. The Hummingbird encryption and decryption cores can encrypt and decrypt a 256-bit message block with 14 clock cycles. 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[1,6] proposed 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.

III. PROPOSED WORK

A novel ultralightweight cryptographic algorithm referred to as a hummingbird is proposed for resourse constrained devices. The design of the hummingbird cryptographic algorithm is motivated by the well known Enigma machine taking into account both security and efficiency[1]. Different from existing lightweight cryptographic primitives which are either block ciphers or stream ciphers, the design of Hummingbird 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 RFID applications.

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*.

Both initialization and encryption consist of four 16-bit block ciphers *Eki* (*i* = 1, 2, 3, 4), four 16-bit internal state registers *RSi* (*i* = 1, 2, 3, 4), and a 16-stage Linear Shift Feedback Register (LFSR). Moreover, the 256-bit secret key *K* is divided into four 64-bit subkeys *k*1, *k*2, *k*3 and *k*4 which are used in the four block ciphers, respectively. After a system initialization process as shown in Figure 1(a), a 16-bit plaintext block *PTi* is encrypted by passing four identical block ciphers *Eki* (.) (*i* = 1, 2, 3, 4) in a consecutive manner, each of which is a typical substitution-permutation(SP) network with 16-bit block size and 64-bit key. The block cipher consists of four regular rounds and a final round. The substitution layer is composed of four S-boxes with 4-bit inputs and 4-bit outputs.

The permutation layer in the 16-bit block cipher is given by the linear transform

$L : \{0, 1\}16 \rightarrow \{0, 1\}16$ defined as follows: $L(m) = m \boxplus (m \_ 6) \boxplus (m \_ 10)$,

where $m = (m0, m1, \cdots, m15)$ is a 16-bit data block.

To further reduce the consumption of the area and power of Hummingbird in hardware implementations, four S-boxes used in Hummingbird can be replaced by a single S-box, which is repeated four times in the 16-bit block cipher.

Hummingbird algorithm:

Hummingbird is neither a block cipher nor a stream cipher, but a rotor machine equipped with novel rotor-stepping rules. The design of Hummingbird 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. The steps performed are:

1) 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.

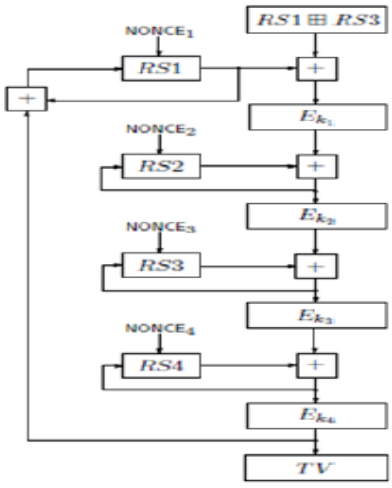

Fig. 1 Initialization Process

2) Encryption Process:

The overall structure of the Hummingbird encryption algorithm is depicted in Figure (2) below. 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.

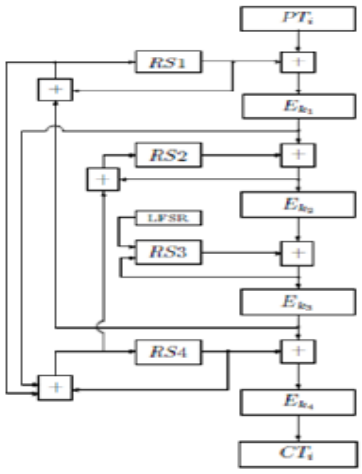

Fig. 2 Encryption Process

3)Decryption Process:

The overall structure of the Hummingbird decryption algorithm is illustrated in Figure (3) below. The decryption process follows the similar pattern as the encryption.

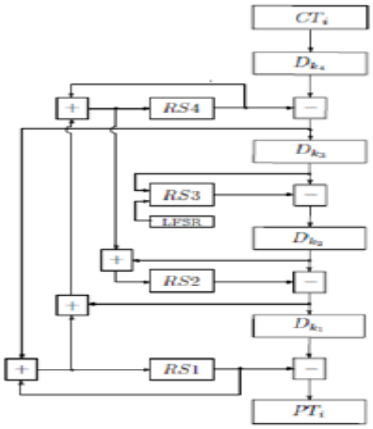

Fig. 3 Decryption Process

4)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 substitution permutation (SP) network with 16-bit block size and 64-bit key as shown in the following figure (4). 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 *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 type S-boxes [1] with additional properties which can ensure that the 16-bit block cipher is resistant to linear and differential attacks as well as interpolation attack[9].

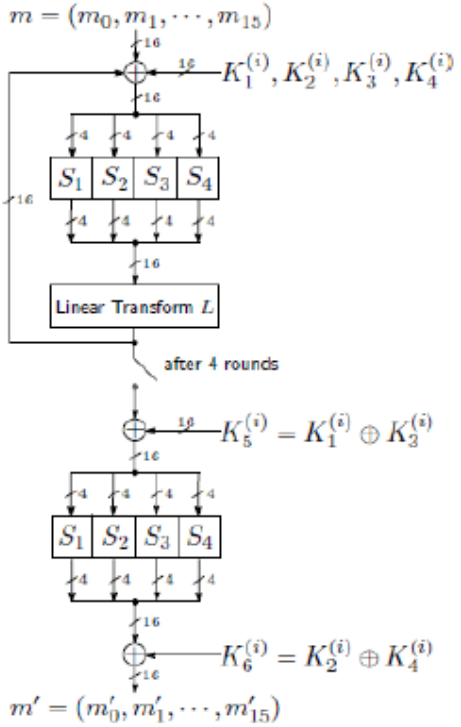

Fig. 4 16-bit Block Cipher

For the implementation of Hummingbird cryptographic algorithm the feasible and available platforms are FPGA, ModelSim. FPGA configuration is specified using a hardware descriptive language. Verilog is the hardware description language used for designing as well as simulation purpose. The hardware design of hummingbird on FPGA is shown using hardware description language as verilog via Quartus II. Virtual model of hardware is simulated via ModelSim. Simulator synthesized using Quatus II. Since we are emphasizing on the hardware implementation of the hummingbird cryptographic so the FPGA is the hardware platform selected depending on the application needs and constraints. After simulation and synthesis the next step is place and route which provides the hardware design for the proposed hummingbird cryptographic algorithm.

In this proposed architecture design of Hummingbird Cryptographic Algorithm we are introducing the concept of pipelining. Pipelining technique reduces the number of clock cycles to get the cipher text thus giving the advantage of increasing the throughput. We will also analyse the no of clock cycles, power requirement, throughput of the Hummingbird

Cryptographic Algorithm also we will give the comparision of Power requirement, area requirement, Input Output delay, number of clock cycles of the Hummingbird cryptographic algorithm with the existing FPGA implementation of the block ciphers XTEA[10],AES,SEA[7].

CONCLUSION

In future, communicating the data for resourse constrained devices through the network is very much important. There are many cryptographic algorithms to achieve this task Hummingbird Algorithm is one of those Algorithm which uses ultralightweight Cryptography for the resourse constrained devices.The additional advantage of hummingbird algorithm is its four round block cipher with feedback operations, which allow to use internal cipher blocks chaining. Compared to other lightweight FPGA implementations of block ciphers XTEA, ICEBERG, SEA and AES, Hummingbird can achieve larger throughput with smaller area requirement using the concept of pipelining.

REFERENCES

[1]Xinxin Fan; Guang Gong; Lauffenburger, Hicks,"FPGA implementations of the Hummingbird cryptographic algorithm", 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp.48-51, 13-14 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 Journal of Computer Applications(0975-8887) Volume 92-No.16,April 2014,42-47.

[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-April 2013, 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 Cryptography: Underlying Principles and Approaches" International Journal of Computer Theory and Engineering, Vol. 3, No. 4, August 2011

[7]F. Mace, F.-X. Standaert, and J.-J. Quisquater, "FPGA Implemenation(s) of a Scalable Encryption Algorithm", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 16, no. 2, pp. 212-216, 2008.

[8]T. Eisebarth, S.Kumar, C. Paar, A. Poscmann, and L.Uhsadel. "A Survey of Lightweight Cryptography Implementations",IEEE Design & Test of Computers.vol.24.n0. 6.,pp 56.

[9]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 on Financial Cryptography and Data Security - FC 2010, 2010.

[10]J.-P.Kaps,"Chai Tea, Cryptographic Hardware Implemenations of xTEA", The 9th International Conference on Cryptology in India - INDOCRYPT 2008, LNCS 5356, pp. 363-375, 2008.