International Journal of Technology and Engineering System (IJTES)
Vol 8. No.2 –2016 Pp. 169-176
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

FPGA IMPLEMENTATION OF PIPELINED ARCHITECTURE IN HUMMINGBIRD ALGORITHM FOR REDUCING AREA AND HIGH THROUGHPUT

K. R. Velmurugan,
M.Tech Communication System,
PRIST UNIVERSITY, THANJAVUR.

Mr. S.Balakrishnan,M.Tech,
Asst. proof, ECE Department
PRIST UNIVERSITY, THANJAVUR.

Abstract—Hummingbird is a Lightweight Authenticated Cryptographic Encryption Algorithm. This light weight cryptographic algorithm is suitable for resource constrained devices like RFID tags, Smart cards and wireless sensors. The key issue of designing this cryptographic algorithm is to deal with the trade off among security, cost and performance and find an optimal cost-performance ratio. This paper is an attempt to find out an efficient hardware implementation of Hummingbird Cryptographic algorithm to get improved security and improved throughput by adding Hash functions. In this paper, we have implemented an encryption and decryption core in Spartan 3E and have compared the results with the existing lightweight cryptographic algorithms. The experimental results shows that this algorithm has higher security and throughput with improved area than the existing algorithms.

Keywords—*Lightweight Cryptography, FPGA Implementation, Security analysis, Mutual authentication protocol.*

I. INTRODUCTION

The importance of RFID tags, smart cards, wireless sensor networks are increasing in our daily life. However the major problem that prevailing now is the lack of information security where the private data can be accessed by the unauthorized person. To protect our privacy, the demands for cryptographic algorithm embedded into these devices are raise. Due to the large area required for AES [I], the most widely used algorithm, it cannot be used for highly constrained environment such as RFID tags, smart cards etc. For that a new research area is put forward called light weight cryptography [2] to obtain the trade-off among privacy, performance and cost. Recently, an ultra-light weight cryptographic algorithm called hummingbird algorithm [3] has been proposed for resource constrained applications. Hummingbird is a combination of both block cipher and stream cipher along with a rotor machine equipped with novel rotor stepping rules.

The design of Hummingbird has a 256 bit key and it takes 16 bit block data and performs the encryption in stream wise with 80-bit internal state. The model is considered as a hybrid model. The hybrid model can provide the designed security with small block size. Therefore it can meet the stringent response time and power consumption requirements for the light weight resource constrained cryptographic applications like RFID tags, Smart cards. The Hummingbird's encryption uses the principle of classic rotor machine, which will perform the substitution and permutation operations with minimal processing requirements and power usage. This algorithm can be implemented by using small number of gates, logic signals and cipher text can be produced in a minimum number of clock signals. Hummingbird is neither a block cipher nor a stream cipher, it's a hybrid structure of both. It will take a block of data

and converted into cipher text in stream wise. So this hummingbird algorithm is resistant to most of the common attacks like structural attack, birthday attack, algebraic attack etc. It can be implemented in variety range of software and hardware platforms with higher efficiency and security. Recently this hummingbird cryptographic algorithm is using in twitter to improve the privacy [4].

The Hummingbird cryptographic algorithm is initially designed on a zero power 4 bit MARC4 microprocessor [5] and 16 bit MSP 430 microcontroller from Texas Instruments [6] and compared the results with another light-weight algorithm PRESENT on similar platforms. The results show that this algorithm can achieve 4.7 times faster throughput than other lightweight algorithms. After that the first efficient hardware implementation of hummingbird cryptographic algorithm using FPGA is proposed in [3],that can encrypt or decrypt a 16 bit message signal within 4 clock cycles after an initialization of 20 clock cycles. Based on the memory blocks embedded in Spartan 3 E, an enhanced implementation of the same in FPGA with co- processor approach is proposed in [7] by using serialized data processing principles. In 2010, a mutual authentication protocol for RFID tags using hummingbird cryptographic algorithm was proposed [8] with smallest tag response time of 15.625/1s.

A throughput and area oriented hummingbird design for FPGA [9] was designed with loop unrolled and round based structure respectively. A UHF RFID tag baseband with the hummingbird cryptographic engine was designed in for secured UHF baseband [10] by using the SMIC 0.13 /1m tech. Finally a remedying structure for hummingbird cryptographic algorithm was designed algorithm by adding two cyclic shifts to avoid the Saarinen attack. All these designs prove that the hununingbird algorithm has high efficiency and throughput across different platforms.

This paper proposed a new FPGA based architecture of hummingbird cryptography with key authentication and improved security by adding hash functions with improved throughput by using loop unrolled architecture with Boolean function representation. The rest of this paper is organized as follows. Section II gives a brief explanation of the Hummingbird cryptographic algorithm in RFID application. Subsequently, in Section III, the FPGA implementation of hummingbird algorithm by using hash functions for RFID applications are described. In Section IV the simulation and implementation results are presented and the results are compared with other lightweight block cipher implementations on the similar FPGA platforms. Finally, Section V provide conclusion of our work experiment.

II. THE HUMMINGBIRD CRYPTOGRAPHIC ALGORITHM IN RFID APPLICATIONS

A. RADIO FREQUENCY IDENTIFICAYION (RFID)

Radio Frequency Identification (RFID) is a rapidly developing technology used for main stream application like automatic object identification, contactless payments etc. In this system, each object is labeled with a responder called RFID tags support a larger set of unique IDs than bar codes which composed of tiny integrated circuits, which will responds to the radio frequency queries from a transceiver called a RFID reader. The RFID tag having circuits which will store the information about the object and processing the identification, as well as a radio frequency antenna for wireless data transmission. The RFID systems will identify an object without physical contact which will reduce the labor, eliminates human errors.

But,the radio communication between RFID tags and the reader raise a number of security issues. For example,RFID systems do not conduct the mutual authentication between RFID reader and tags. So it is,very easy for an un-authorized person to hack the reader or tag to obtain the sensitive information. It may violate the owner's privacy. And also, many possible security threats arises due to the unprotected communication between the RFID tags and the reader.

B. HUMMINGBIRD MUTUAL AUTHENTICATION PROTOCOL

The trust relationship between RFID tags and readers based on the high efficient hummingbird cryptographic is established using

hummingbird mutual authentication protocol. For a secured RFID system, the reader can determine the correct key that is communicating with a tag without exposing the tags identity. In this private identification protocol, the reader initially sends a QUERY signal with a 16 bit SESSION ID as input. After receiving the QUERY, the tag will generate four 16 bit random vectors that will be used for initializing the four status registers. After initialization, it will take RS 1 1\ RS3 message data as input, encrypt it three times and generate three cipher texts CTO,CTl,CT2 as tag indicators. Then the tag will transmit these three cipher texts together with the initialized vectors to the reader. With the key present, the reader can do the encryption and generate three cipher texts and the same will be compared with the three tag indicators. If it matches, then the tag will accept otherwise reject and move for the next tag. The mutual authentication protocol is shown in Fig 1 [8].

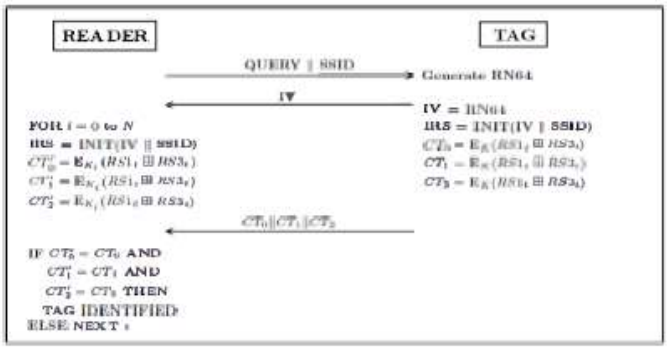


Fig. I Existing Hummingbird Mutual Authentication Protocol

Consider a RFID system with one billion tags, a tag generates three indicators. So, for one billion tags it requires too many indicators (nearly three billion). It takes much area to store these indicators for further comparison. To reduce the area conswnption, this paper proposes a new architecture by using hash algorithm for simplified and secured mutual authentication. It will convert the key into hashes for key security and compare with the hashed keys present in the look up tables for secured key authentication.

III. FPGA IMPLEMENT A TION OF SECURED HUMMINGBIRD ALGORITHM

A. SECURED AUTHENTICATION PROTOCOL BY USING HASH FUNCTIONS

A cryptographic hash function must be able to with stand against all known cryptographic attacks like pre image attacks, algebraic attacks, birthday attack etc. It is very difficult to fmd two different messages with same hash. This property is called strong collision resistance [3]. Here, only the designer knows what mathematical functions are used inside a hash algorithm

Cryptographic hash functions are used to protect the information integrity and authenticity in wide range of applications. Hash functions are cryptographic algorithm that takes a string of any length message as input and returns a fixed size cryptographic hash value. Note that, even a small changes in the message input will drastically changes the resulting hash output. So there is no two message input who having the same hash value. Hence it is very easy to compute the hash value for a given message signal and cannot regenerate the message signal from a given hash. So it is used only for one way communication like to verify the passwords, digital signatures, message authentication codes(MAC's)etc. It is infeasible to modify the message without changing the hash.

B. IMPROVED DESIGN FOR AUTHENTICATION AND KEY SECURITY

Here is a presentation of a design by adding hash functions with the existing hummingbird cryptographic algorithm for secured key authentication and using key security based on low cost Xilinx FPGA Spartan-3. Hummingbird block cipher having four rounds, it is possible to unroll the round loop that makes the block cipher to complete encryption in one clock cycle to improve the throughput. Since the area requirements increase the throughput-oriented design, it will highly improve the throughput. If we want to reduce the area by, we need to trade off, the throughput by using round

based structure. The block cipher will be complete in four clock cycles when the loop is not unrolled.

A modified hummingbird design by adding hash functions for key authentication and improved security for RFID application is given in Fig 2. Instead of using three cipher texts for a single tag, here the hashed output can be used.

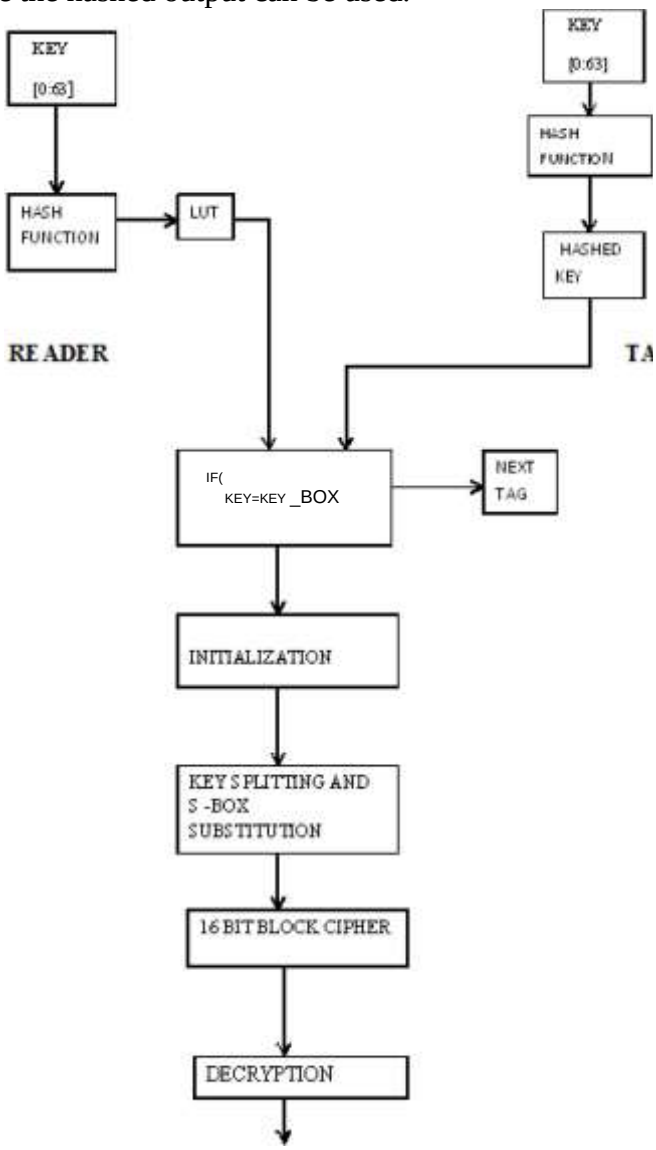


Fig. 2 Flowchart of Secured Hummingbird Mutual Authentication Protocol using Hash Functions.

In the above design the reader first perform the hash algorithm to covert the 64 bit reader key into hashes and sav it in the LUT for future comparison. Similarly in the tag side the tag key will be converted into hashes. Both the hashes should be in equal length. We can compress the hash output by using the compression functions like compressor [3]. Now the hashed tag key will be send to the reader. After receiving the hashed key the reader will compare it with the stored reader keys. If both hash matches, it will accept the tag otherwise will move for the next tag. We can compress the hashed output into any fixed size like 16 bit or 32 bit. Hence the design takes less area to store the hashed key for authentication compared to the existing authentication protocol If key matches, the tag will generate four random initialization vectors with the encrypted data to the reader. After receiving the initialization vectors and the key, the reader will do the initialization process by generating 16 bit cipher text TVO to initialize the LFSR. After initialization, decryption is performed where the encrypted tag data taken as the input. In the throughput oriented design, we unroll the round operation in block cipher to process the 16 bit data in one clock cycle. Instead of using the key directly to improve the security the hashed round key is XORed with the input data and then split into four, each having 4 bits.

C. 64 BIT HASH FUNCTION

The hash function design is user defined. Hashing is not reversible because the input to hash is not one to one mapping. The main security property of hash, famously called collision resistance, demands that it be impossible to find two different inputs which have the same hash value, considerably faster than the birthday attack and only the designer knows the internal structure of a hash function. That is why we can say that the hash function is not broken by an un-authorized person. The block diagram representation of 64 bit hash function is given below in Fig 3(a) and 3(b).

In the above diagram, a 64 bit key is taken as an input. It will be stored in a temporary register called WORD by using a control signal named as RESET. Four 16 bit registers are declared and initialized with random values (only the programmer knows). When the RESET=O, the hash operation
takes place. The 64 bit key will split into four, each having 16 bits. The EH block is nothing but the

hash encryption block where the shifting and round operation will take place according to the random mathematical equation used in that loop. The loop run 16 times and update the Rl, R2,R3,R4 registers within a single clock cycle. The 64 bit hashed key will generate by concatenating four 16 bit hashed sub keys. This 64 bit hashed key will be given as input key for the hwnmingbird initialization block for key authentication. Because of the irreversible hash operations, there is no possibility for an un authorized person to crack the key. So this module will improve the security features.

The top level design for secured hummingbird cryptographic algorithm for encryption is given in Fig 4.

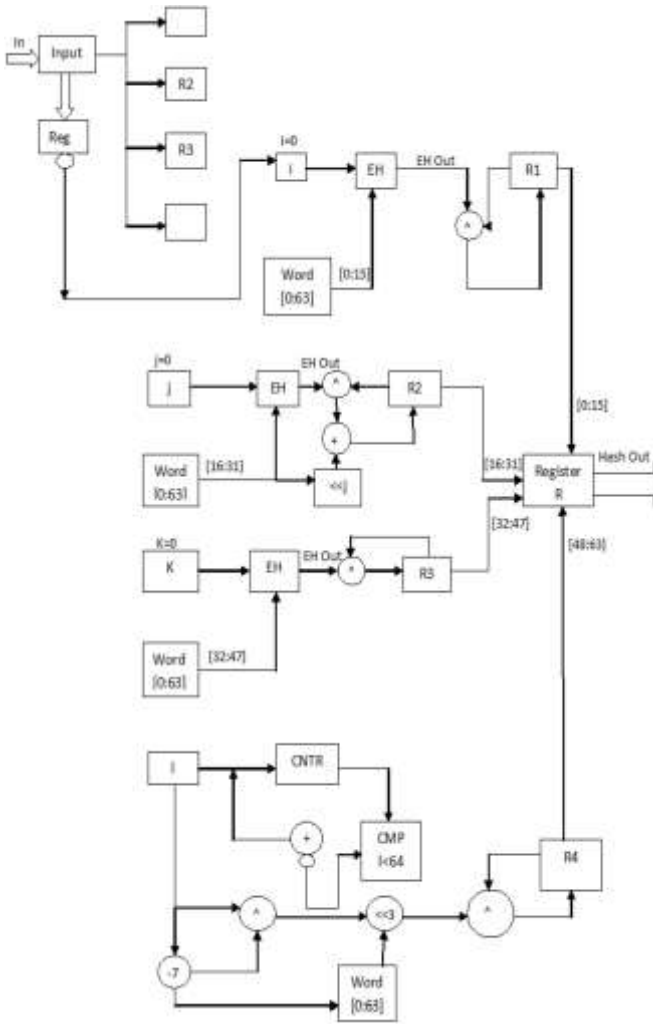


Fig. 3(a) Proposed Hash Function Main Module

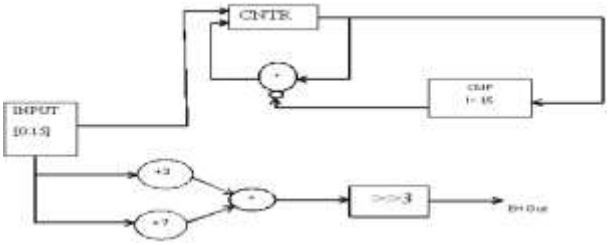
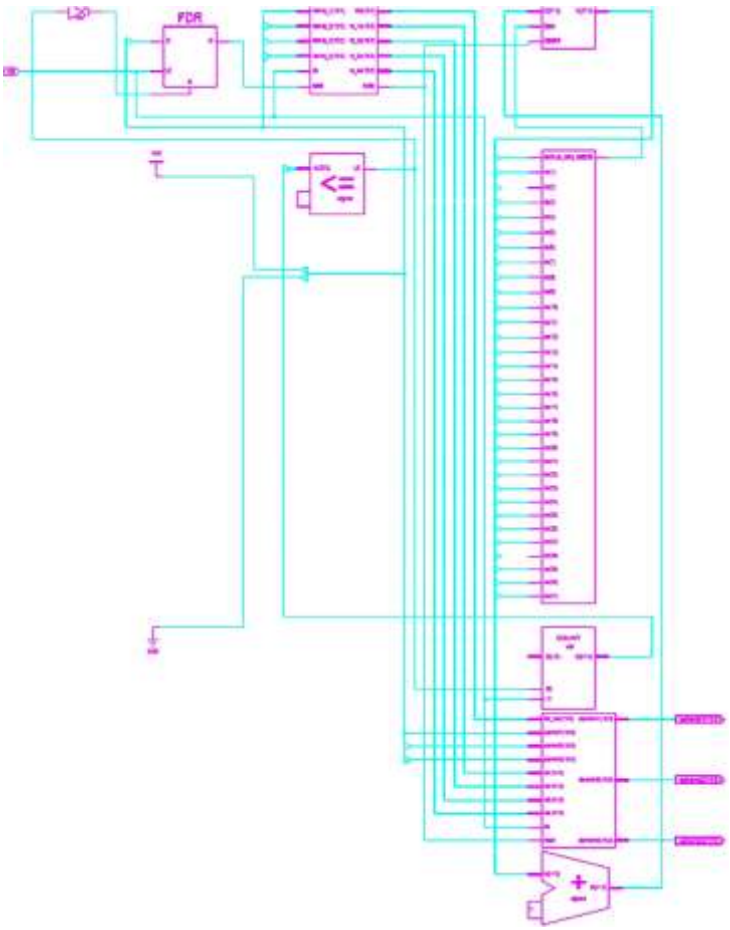


Fig. 3(b) Hash Encryption(EH) Module

IV. SIMULATION RESULTS

The improved hummingbird cryptographic algorithm has been coded by using Verilog HDL. All the results are simulated and synthesized by using Xilinx ISE 1O.l.

The simulation result of hash function is shown in Fig 5. Fig 6 shows the simulation result of secured hummingbird encryption using hash functions



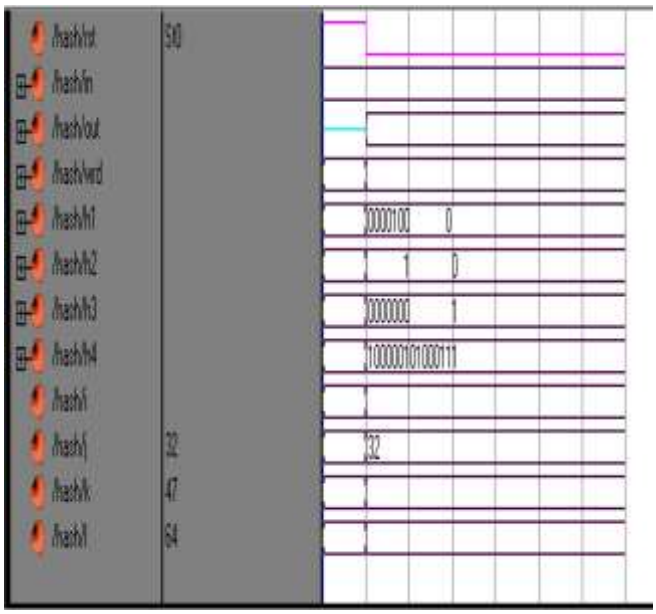


Fig. 5 Simulation results of Hash Function

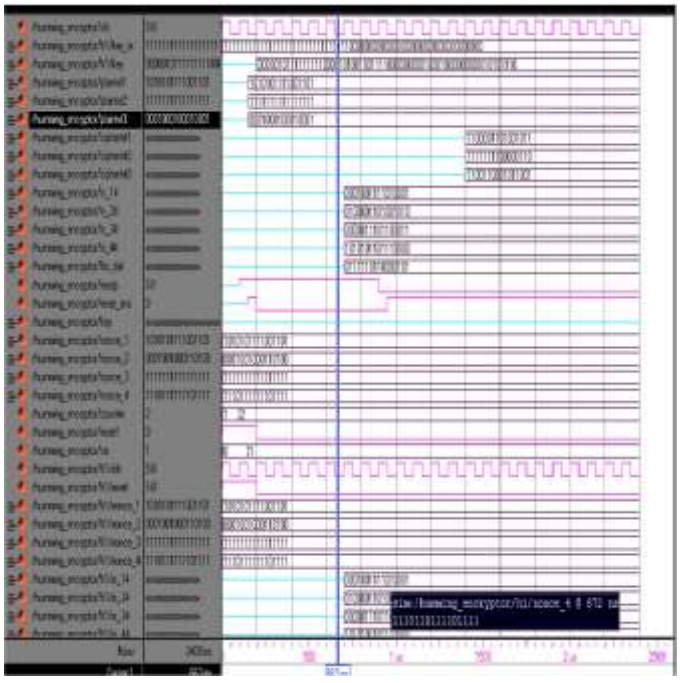


Fig. 6 Simulation results Secured Hummingbird Encryption using Hash

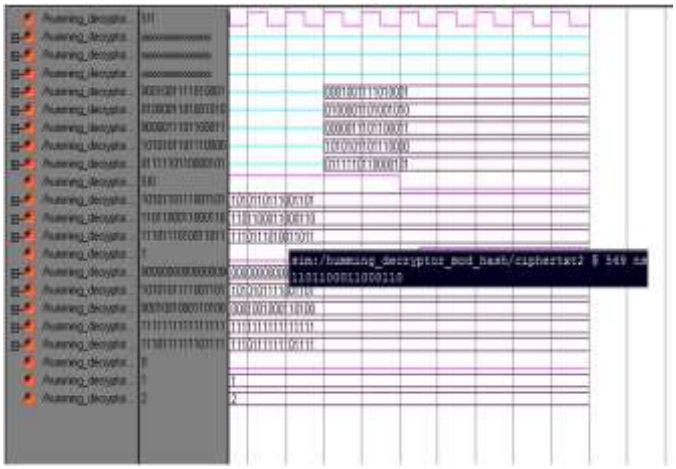


Fig. 7 Simulation results of Hummingbird Decryption without a matching

.

In this design, because of key authentication the tag does not reveal the data stored inside it. If the hashed key is not matches with the hashed reader key that stored in the Look Up Table, it will not perform the decryption operation. While in key authentication the comparison process will take place in the initialization block. If the keys are not matches, then the decryption block will generate an invalid output without revealing the exact data stored. The simulation result of decryption after validation process is given in Fig 7 and Fig 8. As shown in Fig 7, a given key that does not match with the stored key, cipher texts as input but do not regenerate the plaintext. The decrypted output after the successful key authentication is given in Fig 8.

Fig. 8 Simulation results of Hummingbird Decryption after matching the key.

V.COMPARISON

The designed implementation in Verilog based on low ost Xilinx FPGA Spartan 3 XC3S200 in package FT256 and with speed grade -5 and took the area and power requirements by using Design Compiler. Xilinx ISE 6.4 was employed for implementation. We have compared our results with the other designs [3] [9]. First FPGA implementation for hununingbird cryptographic algorithm proposed [5] is an efficient architecture which cost only four 16 bit adders. However the top level design consists of too many multiplexers. How throughput and small area is provided in the hardware implementation is proposed [3] which takes 16 clock cycles for a block plain text. Throughput is much smaller and costs very few areas in [7], but requires additional internal memories in FPGA.

Due to the introduction of co-processors in the design which is implemented by FPGA the internal block memories are used as instruction memory and register files. But it cannot be considered as hardware implementation because instructions are stored in the block memory of FPGA. The loop unrolled throughput and round based area oriented designs [9] for improved throughput and area respectively. It doesn't improve the throughput much and a chosen- IV messages attack on hummingbird is given by FSE'II [4]. So this modified design improves the security by using hash functions which will not affected by all the known attacks. The number of slices occupied is used to decide the area requirement in former designs. But it is very hard to compare the area requirements. The efficiency of the whole design is defmed as the ratio of slice number to throughput. The area, gate count, memory requirements, throughput are given in Table I and comparison results are shown in Table II. The results of the design and comparison with existed methods are shown in Table I, II. The area requirements are found out from Xilinx ISE and the gate count, power calculated from Design Compiler after power optimization by using L VT techniques.

TABLE I RESLUTS OF SECURED HUMMINGBIRD DESIGN AND COMPARISON WITH OTHER METHODS IN LITERATURE

	Area (Slices)	Memory (Blocks)	Frequency (MHz)	Throughput (Mbps)
X. Fan [3]	273	0	40.1	160.4
Bio Min, Ray C.C [6]	229	0	39.8	157.6
Proposed Hummingbird design with secured key authentication using hash	485	0	42.840	171.36

The results shows that the algorithm with an extra module of hashing can promote higher throughput and frequency than
X.Fan and Bio Min by with the cost of higher resources. The additional hash module that used for key authentication and security will take a total cell area of 3659.673666 with power 184.637611

TABLE II SECURED HUMMINGBIRD DESIGN COMPARISON WITH THE DESIGN WITHOUT HASH FUNCTION

	Gate Count	Power (µw)	Area (Slices)	Throughput (Mbps)	Efficiency (Mbps /Sec)
Hummingbird algorithm without Key authentication	34266.938539	482.3713	467	172.73	0.3698
Proposed Hummingbird algorithm using Hash functions.	45138.406945	326.3007	485	171.36	0.3533

The above table shows the comparison of the proposed secured hummingbird design with key authentication using hash function with the design without hash module. The
design shows that the algorithm having improved security and throughput with a cost of small

increase in area. This design has less power consumption than the previous design with an efficiency of 0.3533.

VI. CONCLUSIONS

This paper presents an improved architecture of hummingbird cryptography by key authentication using hash functions. We can obtain an optimized throughput for high throughput requirements with the additional cost of larger area. It achieves with no memory blocks occupation. These results show that the proposed designs have higher throughput, less power consumption, high security than existed methods.

For the future work, we would evaluate the security performance of the hummingbird cryptography as it may be directly influenced by the random initial value of the four state registers, which are read directly from outside. To understand how to generate these initial values on hardware is also a topic of interact to future.

ACKNOWLEDGMENT

The authors knowledge the help of Mr. C. Babu, Assistant Professor, Dept. of Electronics and Communication, Amrita School of Engineering Bangalore, Amrita University for introducing the various aspects of digital design.

REFERENCES

[1] P. Chodowiec and K. Gaj, "Very Compact FPGA Implementationof the AES Algorithm," in Cryptographic Hardware and Embedded Systems - CHES 2003. vol. 2779, C. Walter, c,:. Koy, and C. Paar, Eds., ed: Springer Berlin Heidelberg, 2003, pp. 319- 333.

[2] R. RajaRaja and D. Pavithra, "Implementation of hardware efficient light weight encryption method," in Communications and Signal Processing (lCCSP), 20i3 International Conference on, 2013, pp. 191-195.

[3] F. Xinxin, G. Guang, K. Lauffenburger, and T. Hicks, "FPGA implementations of the Hummingbird cryptographic algorithm," in Hardware-Oriented Security and Trust (HOST), 2010 iEEE International Symposium on, 2010, pp. 48-51.

[4] E. De Cristofaro, C. Soriente, G. Tsudik, and A. Williams, "Hummingbird: Privacy at the Time of Twitter," in Security and Privacy (SP), 2012 IEEE Symposium on, 2012, pp. 285-299.

[5] 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, pp. 1-7

[6] D. Engels, X. Fan, G. Gong, H. Hu, and E. Smith, "Hummingbird: Ultra-Lightweight Cryptography for ResourceConstrained Devices," in Financial Cryptography and Data Security. vol. 6054, R. Sion, R. Curtmola, S. Dietrich, A. Kiayias, J. Miret, K. Sako, et al., Eds., ed: Springer Berlin Heidelberg, 2010, pp. 3-18.

[7] T. San and N. At, "Compact Hardware Architecture for Hummingbird Cryptographic Algorithm," in Field Programmable Logic and Applications (FPL), 201i International Conference on, 2011, pp. 376-381.

[8] X. F. Daniel Engels, Guang Gong, Honggang Hu, Eric M. Smith, "Ultra-Lightweight Cryptography for Low-Cost RFlD Tags:Hummingbird Algorithm and Protocol," FC'IO Proceedings of the 14th international conference on Financial cryptograpy and data security,Springer-Verlag Berlin, Heidelberg ©20JO, vol. ISBN:3-642-14991-X 978-3-642-14991-7, pp. 3-18 2010.

[9] M. Biao, R. C. C. Cheung, and H. Yan, "FPGA-based highthroughput and area-efficient architectures of the Hummingbird cryptography," in iECON 2011 - 37th Annual Conference on iEEE industrial Electronics Society, 20 II, pp. 3998-4002.

[10] X. Mengqin, S. Xiang, W. Junyu, and J. Crop, "Design of a UHF RFID tag baseband with the hummingbird cryptographic engine," In ASiC (ASiCON), 2011 iEEE 9th international Conference on, 2011, pp. 800-803.