Turkish Journal of Computer and Mathematics Education *Vol.12 No.14 (2021), 5612-5622*

Research Article

Remedying the Hummingbird Cryptographic Algorithm

B.Vishwa Teja[a], R.G.Kalyan Sreenivas[b]

[a] Student, Electronics and Communications Engineering, Bachelor of Technology, Vellore Institute of Technology
E-mail: bvtvishwa@gmail.com
[b]Student, Electronics and Communications Engineering, Bachelor of Technology, Vellore Institute of Technology
E-mail:kalyansreenivas3970@gmail.com

Abstract: In the present current life, there is evermore care is expected to ensure data. Therefore, it is important to consider the cryptographic algorithms. The different cryptographic algorithms like AES, DES have been neglected to meet the necessities of low-level devices particularly in the control systems, and this prompts the advancement of super lightweight cryptography. Lightweight cryptography is the new region in the field of cryptography. These lightweight cryptographic algorithms attempt to decrease area and power requirements, with less an ideal opportunity for handling. This prompts minimal expense execution. Hummingbird cryptography is one such strategy.

Different lightweight cryptographic algorithms are examined momentarily, these algorithms utilize either block cipher or stream cipher yet the hummingbird is the uncommon lightweight cryptography since it utilizes the highlights of both block and stream cipher. Accordingly, it gives better security to less-resourced devices in examination with the other lightweight algorithms. The hummingbird algorithm is now planned and created on various stages like microcontroller, ASIC execution, and so forth, and it is examined momentarily.

Keywords: low-level devices, lightweight cryptography, hummingbird algorithm

1. Introduction

Data and Information sharing has become a major part of our life because there has been a huge leap of advancements in network and communication technology and due to that fact, there arises a problem of someone else stealing our data and hacking into our systems. To satisfy these requirements number of devices were structured and created with the primary objective of shielding data present in the computer from hackers. These facilities are generally called computer security. Cryptography is a procedure that makes data not lucid by unapproved people, by the utilization of encoding strategy. A portion of the use of cryptography incorporates the security of ATM cards, the utilization of passwords in computers, and electronic trade. Cryptography is gotten from the Greek words 'Kryptos(hidden) and 'Graphos'(written) individually. Cryptography includes two processes. These are encryption and decryption. Encryption changes the plain content over to ciphertext. Decryption converts cipher text to plain text. There are different cryptographic algorithms.

The design of any cryptographic calculation requires the emphasis on three-parameters these are security, cost, and performance. So relying upon a specific application one should design their algorithm accordingly. Normalized algorithms like AES, DES and so forth, neglects to give security in applications like RFID tag, sensor hubs, and smart cards. These algorithms require the help of more resources, lightweight cryptography can be utilized as an option for these applications, which gives better security contrasted with normalized cryptographic algorithms. This gives inspiration for the creator to move towards lightweight cryptographic algorithms. Hummingbird cryptography is one of the lightweight cryptographic algorithms. The plan of Hummingbird cryptography is roused by the enigma machine (**Daniel Engels, Xinxin Fan**). It is found that it uses small-sized blocks and able to withstand cryptographic attacks like linear and differential cryptanalysis (**F. Chabaud and S. Vaudenay**). In this paper hummingbird, the cryptographic algorithm is thought of and attempts to change it to diminish the area and enhanced performance without influencing the security of the algorithm. Lightweight cryptography is another exploration region. These calculations attempt to have lower area, lower power, low cost, and less processing time. Hummingbird cryptography is one such technique. Hummingbird cryptography can be considered as an FSM (Finite State Machine) in light of the incorporation of highlights of both block cipher and stream cipher. Subsequently straightforward and prompts better security for control framework application.

2.Significance Of The Study

As we understood that lightweight cryptography is mainly focused on providing security to less-resourced devices like smart cards, RFID tags, and sensor nodes. So while developing the architecture of lightweight cryptography it should be noted that the designed architecture should not increase the area of the system where it is to be applied. Usually, the implementation can be done by using any one of the following methods: parallel (loop unrolled), round wise, or serial‖ architectural options.

A parallel block cipher algorithmic implementation takes only one clock cycle for many rounds of operation of block cipher both in encryption /decryption processes. Pipelining concept may be employed; this can be done by using registers. This clock frequency of operation is further increased. So the parallel approach is better where higher performance is required. But for algorithms designed for RFID application should not follow this, because RFID requires higher area and power, instead of higher throughput. For the second approach i.e. round-wise development of architecture, one clock cycle is required for each round of cipher used, regardless of either we have used block cipher or stream cipher. With a stream cipher, this approach results in a low area and low power requirements. To achieve low power and reduced area, serial implementation is best suited. In this case, a single clock cycle completes only a fraction of the round operation. So more than one clock cycle is required to complete a given round operation, resulting in less performance. So depending on the application, the architectural option of the algorithm is to be chosen.

3.Review Of Related Studies

Based on the proposed algorithm by (**Biao Min, R. C. C. Cheung and Yan Han**) Hummingbird uses a 256-bit key and takes a 16-bit wide block of a message as an input. These represent the characteristics of block encryption. This algorithm also uses 4 internal state registers, each of which is 16 bit wide and an LFSR (Linear Feedback Shift Register) of 16bit width. These represent the feature of stream encryption. So it is considered as a combination of a block cipher and stream cipher.

Like other cryptographic algorithms it also includes three processes, they are initialization, encryption, and decryption. The initialization process is the first process to be taken place, where initial values are applied to the state register. These initial values are 16-bit random numbers. In some references, these numbers are referred to as a number at once (noncei). After the completion of the initialization process, encryption starts, where the given plain text is converted to ciphertext by using a particular number of permutation and substitution methods. These permutations and substitutions are designed as a part of the block cipher. The decryption process is the reverse operation of encryption, where we obtain plaintext back from the ciphertext. Usually, initialization and encryption processes follow the same procedure. But the main goal of initialization is to initialize the state registers, while the main intention of encryption is to convert the given plain text or 16-bit block data into another 16-bit block data (ciphertext), which is not readable (understandable) by an unauthorized person. Thus, it is valid to say that initialization and encryption have slight differences. This concept helps us to reduce the area requirements by resource sharing.

3.1 Block cipher

This algorithm uses a 256-bit key, 16-bit block of data, and 4 internal states register each with 16-bit wide. The initialization process aims to get the initial value of LFSR. In the initialization process registers are initialized with some random values of 16 bit wide, named as NONCE(number at once). The output of initialization is named TV. During the encryption process, The required plaintext is given as input data, and then block cipher operation takes place with the updated values of registers obtained from the initialization process. Finally at the end of the encryption process ciphertext (CT) is obtained.Both initialization and encryption processes use 4 block cipher, so each block cipher uses one sub-key.

Figure 1 shows the structure of the block cipher used **(Meng-Qin Xiao, Xiang Shen)**. Each block cipher uses a 64-bit sub-key, obtained from 256-bit wide key. These sub-keys are used by dividing them as 16-bit wide round keys for each round. This key partition is shown on the right side of figure 1.As shown in figure 1 the block cipher consists of 4 regular rounds and a final round. Each of the regular rounds includes a key mixing step, substitution step, and permutation. The final round includes only substitution and key mixing steps. The substitution is done by using S-boxes having 4-bit input and 4-bit output. The key mixing is done using a simple XOR operation. The first round key mixing is performed between round key k1 and input data and the output of previous rounds is used as input data for remaining rounds(2nd,3rd and 4th rounds) with round keys k2,k3, and k4 respectively.

The MUX selects input data during 1st round operation and it selects the output of previous rounds as input data for remaining rounds i.e. for 2nd, 3rd, and 4th rounds. It is done by using the control signal, applied to the

MUX. Here the clock signal is used as the control input. This operation repeats 4 times the final round operation takes place, here the keys used are k_5 and k_6. Key k_5 is the result of the XOR operation of k_1 and k_3, similarly, k_6 is the result of XOR operations of k_2 and k_4. Finally at the end of the final round operation output of the block cipher is obtained (figure 2).

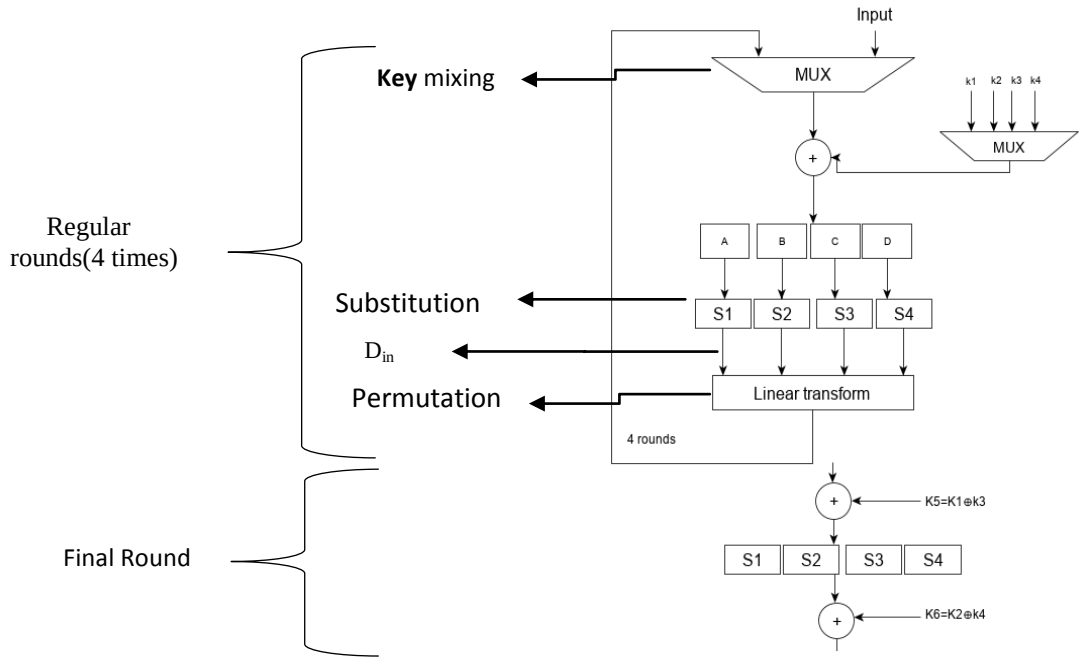

Figure 1: Structure of block cipher

As described above the block cipher involves 4 regular rounds and final round operations.
As described in figure 1, the block cipher includes 4 regular rounds and one final round operation.
Each regular round involves
a)key mixing: It is done by using XOR operation
b) substitution: It is done by using an s-box.
c)permutation: By using linear transform, expressed as

$$D_{out} = D_{in} \text{ XOR } (D_{in}<<6) \text{ XOR } (D_{in}<<10).$$

Here D_{in} is the output of the s-box module, it is given as input to the permutation step and the Dout represents the output of linear transform. It is used as the input to the next round operation as shown in figure 4.2.
The final round involves a)key mixing and b) substitution.

The key used in the AR, TE, and hybrid structure encryption algorithms is 256 bit wide, these algorithms involve 4-block cipher operations so the key is then divided into 64-bit wide subkeys and each block cipher operation uses one sub-key. For block cipher operation the given sub-key is further divided into 4 round keys, each of which is 16 bit wide.

3.2Throughput-Enhanced (TE) Design

Throughput enhanced design (**Biao Min, Ray C.C.Cheung, Yan Han)**involves unrolling the round operation in the block cipher. It takes only one clock cycle for each block cipher operation because all 4 rounds are completed in a single clock cycle. So it requires 4 clock cycles for initialization and another 4 clock cycles for encryption. So after the end of 8 clock cycles, the output, i.e. ciphertext is obtained.

3.3 Area-reduced (AR) Design*:*

Compared to throughput-enhanced design **(Biao Min, Ray C.C.Cheung, Yan Han]** which requires a single clock cycle for each block cipher operation, the AR method completes one block cipher operation in 4 clock cycles, so it makes the throughput lower. However, the area requirement is smaller due to the resource sharing of four rounds in the block cipher. In this case, each block cipher requires 4 clock cycles. As initialization and encryption use 4 block cipher, each of these processes requires 16 clock cycles, so after the end of the 32nd clock cycle, the ciphertext is obtained.

4.Objectives Of The Study

- To develop a new design that is the hybrid structure of AR and TE.
- Verilog language and Xilinx ISE design tool used for simulation.
- To compare the performance of AR, TE, and hybrid of AR - TE designs in terms of area and speed

5. Proposed hybrid design

From the above explanation, it is clear that TE design requires more area than AR design, while AR design has less throughput than TE design. It shows that there is a need to design a hybrid structure of TE and AR, such that it consumes less area than TE design and gives more throughput than AR designs without affecting the security. The structure of these two designs is implemented in Verilog and results are verified. The hybrid of these two structures is designed and compared with AR and TE structures.

The design is developed by using the block cipher shown in figure 2. This block cipher is designed by using the feature of block ciphers used in AR and TE designs i.e. it is the modified version of both of the block ciphers. Figure 2 shows that the hybrid design uses substitution, permutation operations, and key mixing operations, but the way of functioning is different from AR and TE designs.

In this block cipher, only 3 'regular round' operations are rolled, it is the feature obtained from the AR design. Instead of rolling 4 regular rounds operation as in the case of AR design, here only 3 'regular rounds' operations are rolled. The 4th round operation is done serially with the final round operation, this is the feature obtained from TE design. It shows that 3 clock cycles are required to complete this block cipher operation. So it is expected that by using this hybrid cipher, the area should be less than TE design and throughput should be more than AR design. Here substitution is done using new s-boxes designed by using polynomial in Galois field. The key mixing is done using simple XOR operation and permutation is done using linear transformation as used in AR and TE designs.

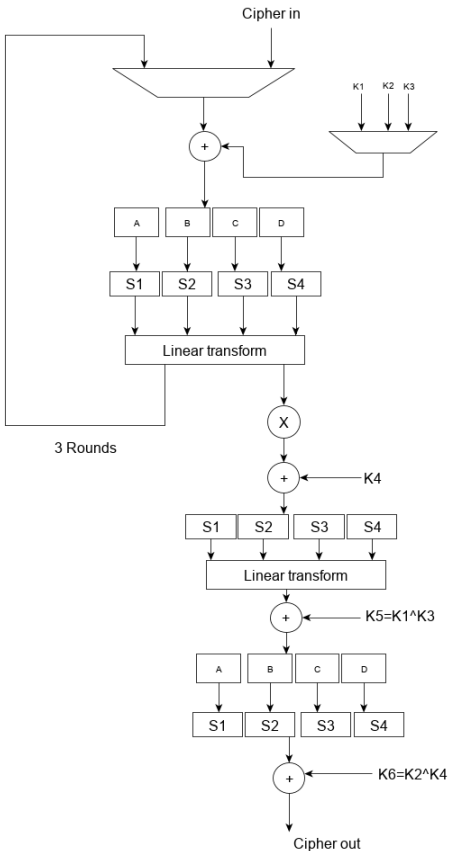

Figure 2: Block cipher of the proposed design

5.1 Top-level Architecture:

The top-level architecture **(Biao Min, Ray C.C.Cheung, Yan Han)** of the proposed hybrid design (Figure 3) is similar to that of AR and TE designs with slight modification. In this, the block cipher used is the proposed one (Figure 2) and substitution is done using new s-boxes generated.

In the top-level architecture RS1, RS2, RS3, RS4 represent 16-bit wide registers. MUX M1, M2, M3, M4, M5 are used to initialize or update the 4 internal state registers. MUX M6 and M7 are used to select registers and data input depending on the number of clock cycles, which is counted by a one-hot counter. Here 'one-hot counter' is used to select data and registers and it is used to select initialization or encryption process depending on the counting value.

Here only one block cipher Ek1 is used, so Ek1 is processed 4 times, to complete all the 4 block cipher operations. So registers are necessary to store the intermediate values between these processes. The hybrid block cipher, described in figure 2, is used in this architecture. This block cipher operation is repeated 4 times for each initialization and encryption process. Here initialization and encryption processes require 12 clock cycles. So finally after the end of 24 clock cycles, the encrypted output is obtained.

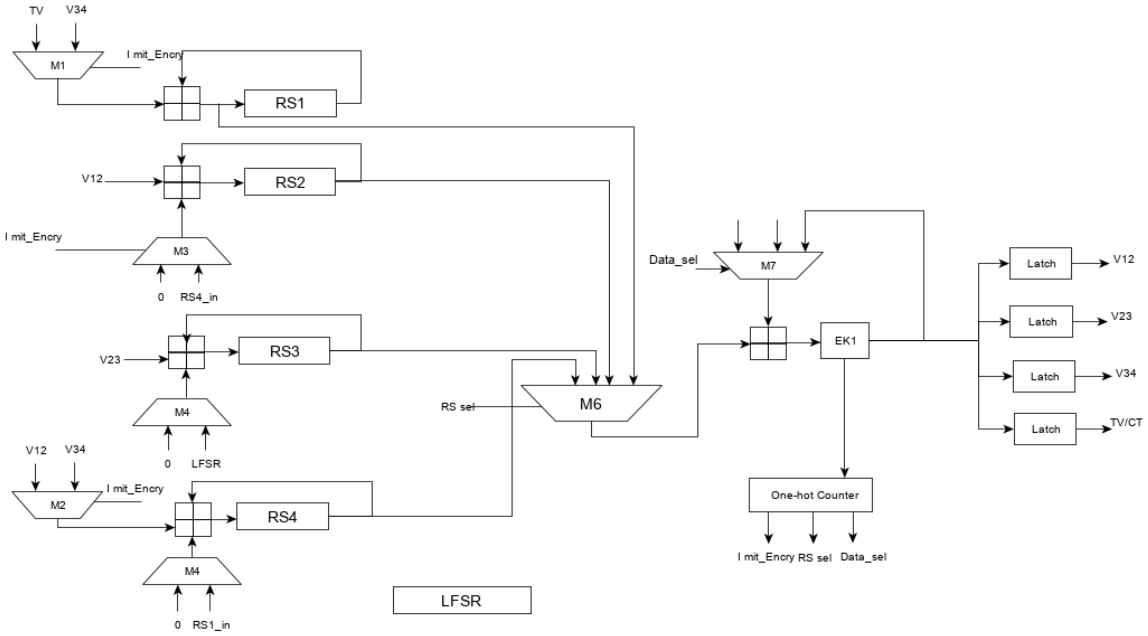

Figure 3: Top-level architecture of proposed design.

5.2 Design of S-box

Table 1: S-box

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S1(x)	0	1	2	4	8	3	6	C	B	5	A	7	E	F	D	9
S2(x)	0	1	2	8	4	9	B	F	7	E	5	A	D	3	6	C

This s-box is used in the initialization and encryption process, i.e. to convert plaintext to ciphertext. In the same way, different s-boxes can be generated using the different irreducible polynomial. Then the s-box which consumes less area is chosen and repeated for the required number of times. Thus area consumption can be reduced.

6. Simulation Results

The performance evaluation of the s-box, block cipher, and encrypted output of area reduced, throughput enhanced and hybrid design was done by describing the functionality of the algorithms in the Xilinx platform using Verilog code.

This paper aims to develop the hybrid design of block cipher that should consume less area than TE design and throughput should be more than AR design. The encryption process implementation of TE, AR, and hybrid designs is considered to compare the performance parameter like throughput, area, number of clock cycles.

Input data be:ff11(hexadecimal)
Sub-key: f00ff00ff00ff00f(hexadecimal)
Resulting in 4 round keys k1, k2, k3, and k4, each having a value f00f.

6.1. Simulation results of encryption process:

The results obtained through simulation for s-box and block ciphers, which are essential parts of the Encryption process and top-level module of encryption for TE, AR, and proposed hybrid design are shown below.Simulation results of the s-box are shown in figure 4. In the simulation waveform of the s-box, for the input 485b, the output obtained is 4737.

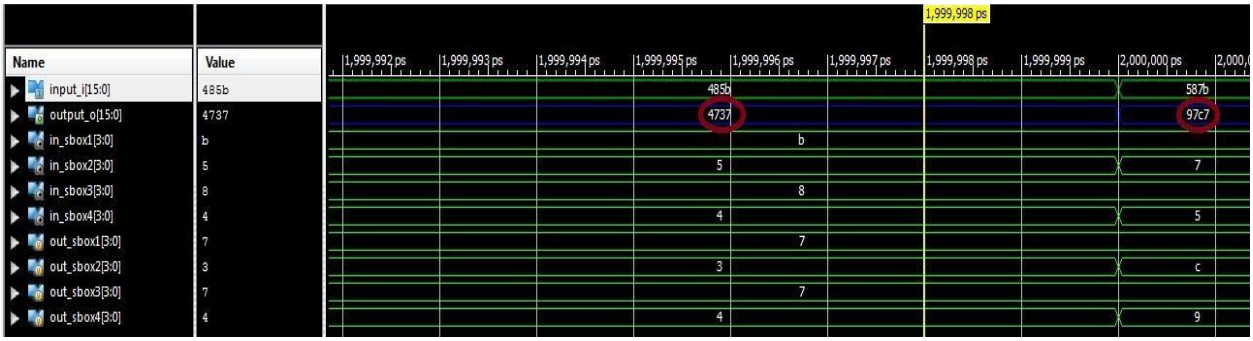

Figure 4: S-box output

Simulation results of an encryption block cipher for TE, AR, and proposed hybrid design are shown in Figures 5,6, and 7 respectively. The output of block cipher obtained for input ff11 is 87c0. The waveform analysis shows that the block cipher operation requires one, four, and three clock cycles for TE, AR, and proposed hybrid designs respectively.

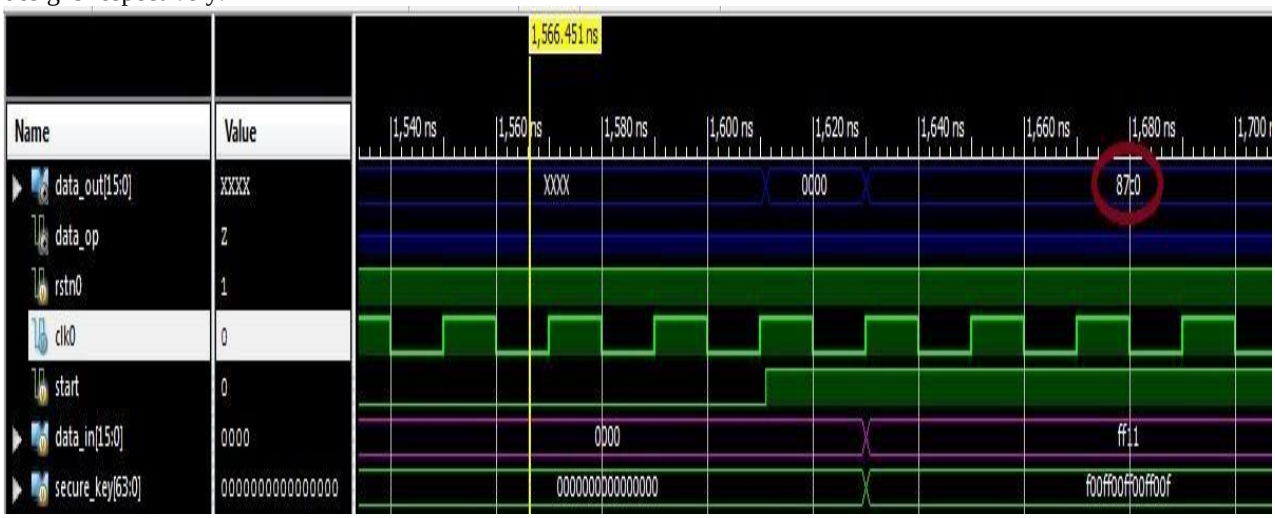

Figure 5: Encryption block cipher waveform of TE design

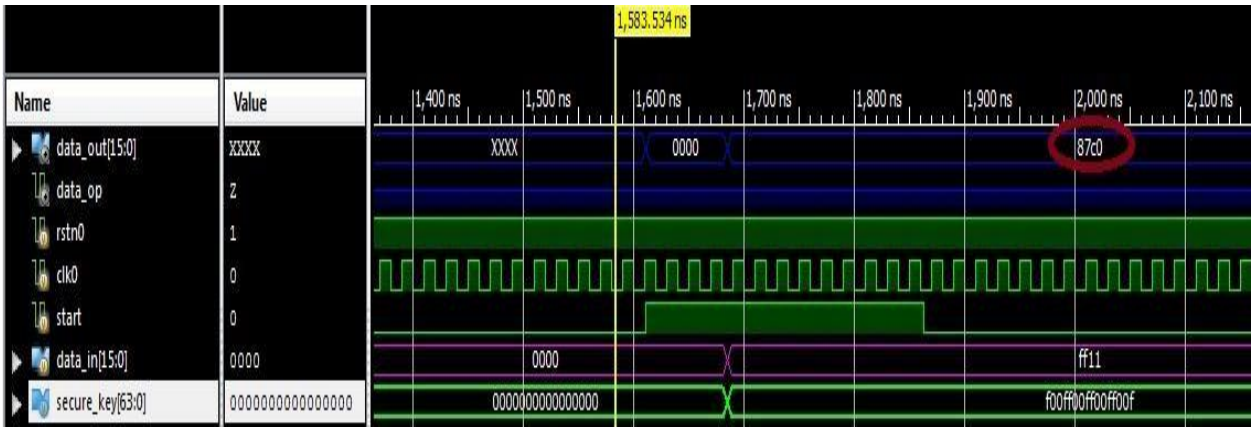

Figure 6: Encryption block cipher waveform of AR design

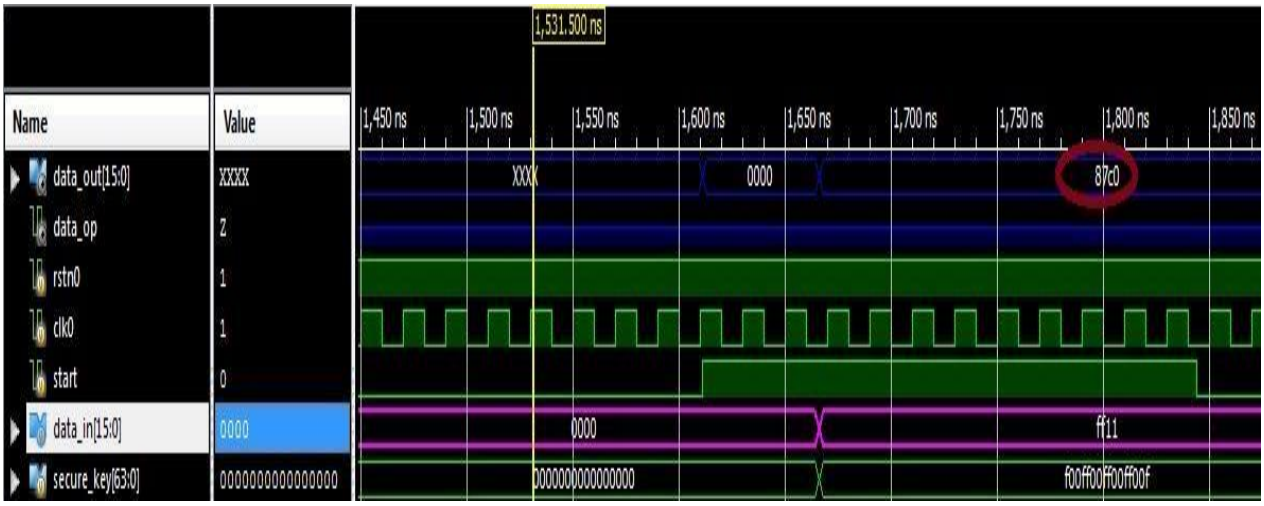

Figure 7: Encryption block cipher waveform of proposed hybrid design

The simulation waveform of encryption of TE design shows that after the 4 clock cycles of start(enable) pin high, the initialization output is obtained.As shown in figure 8 for the given input 4444 after the 4 clock cycles of start pin high the initialization output 4e89 is obtained and encrypted output fb97 is obtained at the end of the next 4 clock cycles. So only 8 clock cycles are enough to get the encrypted output hence it is named as throughput enhanced implementation of hummingbird cryptography.

Figure 8: Output waveform of encryption of throughput enhanced design

The simulation waveform of AR design shows that after the 16 clock cycles of start pin high, the initialization output is obtained As shown in figure 9 for the given input 4444 after the 16 clock cycles of start pin high the initialization output 4e89 is obtained and encrypted output fb97 is obtained at the end of the next 16 clock cycles. So a total of 32 clock cycles are required to get the encrypted output thus it decreases the throughput but area consumption is reduced hence it is named the area-reduced implementation of hummingbird cryptography.

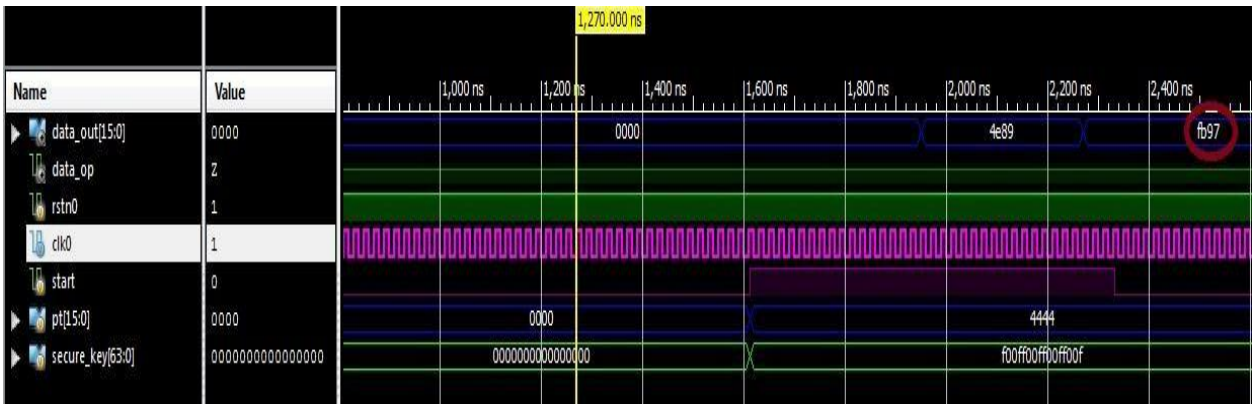

Figure 9: Output waveform of encryption of area reduced design

As shown in figure 10 for the given input 4444 after the 12 clock cycles of start pin high the initialization output 4e89 is obtained and encrypted output fb97 is obtained at the end of the next 12 clock cycles. So a total of 24 clock cycles are required to get the encrypted output, thus it results in less area consumption than TE design and higher throughput than AR designs, hence it is named as a hybrid of TE and AR implementation of hummingbird cryptography.

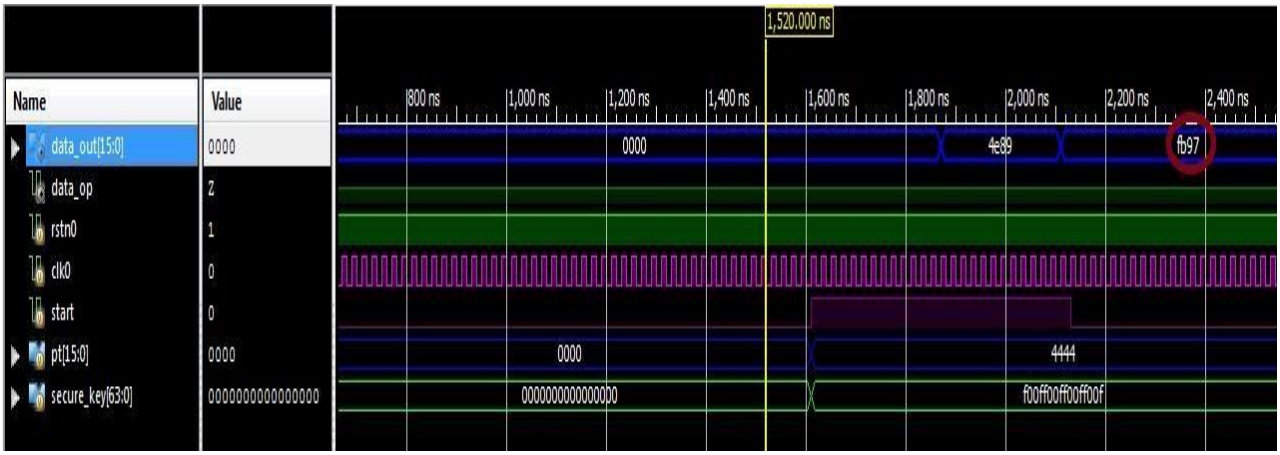

Figure 10:Output waveform of encryption of proposed hybrid design

7.Performance comparison of TE, AR, and Hybrid algorithm:

The design summary of TE, AR, and proposed hybrid design is shown below in tables 2,3, and 4 respectively. The analysis of this table verifies that the number of slice registers used in the modified design is less than that of TE design and slightly more than that of AR design. Other parameters like throughput, number of clock cycles required to get the ciphertext, minimum clock period, and memory usage is tabulated for TE, AR, and hybrid designs in table 5.

Table 2: Design summary of TE design

Device Utilization Summary			
Slice Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	226	54,576	1%
Number used as Flip Flops	134		
Number used as Latches	60		
Number used as Latch-thrus	0		
Number used as AND/OR logics	32		
Number of slice LUTs	1,848	27,288	6%
Number used as logic	1,848	27,288	6%
Number using O6 output only	1,618		
Number using O5 output only	35		
Number using O5 and O6	195		

Table 3: Design summary of AR design

Slice Logic Utilization	**sed**	**Available**	**Utilization**
Number of Slice Registers	172	54,576	1%
Number used as Flip Flops	140		

Number used as Latches	0		
Number used as Latch-thrus	0		
Number used as AND/OR logics	32		
Number of slice LUTs	1,823	27,288	6%
Number used as logic	1,823	27,288	6%
Number using O6 output only	1,540		
Number using O5 output only	35		
Number using O5 and O6	2485		

Table 4:Design summary of proposed hybrid design

Device Utilization Summary			
Number of Slice Registers	174	54,576	1%
Number used as Flip Flops	142		
Number used as Latches	0		
Number used as Latch-thrus	0		
Number used as AND/OR logics	32		
Number of slice LUTs	1,752	27,288	6%
Number used as logic	1,752	27,288	6%
Number using O6 output only	1,464		
Number using O5 output only	37		
Number using O5 and O6	251		

Various parameters like area consumption, throughput, number of clock cycles required to get the ciphertext, frequency of operation are shown in table 5 for area reduced, throughput enhanced, and hybrid designs. The table 4 shows that hybrid design requires less area than TE design and yields better throughput than AR design.

Table 5: Area and throughput comparison of AR, TE, and proposed hybrid design.

Parameter	**AR**	**TE**	**Proposed design**
Frequency(MHz)	29.94	36.18	30.202
Number of slice register	172	226	174
Number of clock cycles	32	8	24
Minimum clock period(ns)	33.401	27.639	33.110
Memory usage(MB)	310	320	311
Throughput(Mb/s)	479	578	483

Table 6 shows the comparison of AR, TE, and hybrid designs in terms of area and throughput. The output waveform analysis of area reduced, throughput enhanced and hybrid design verifies that the number of clock cycles required to get the encrypted output for hybrid design is in between the value of the AR and TE designs. So it verifies that hybrid design is the optimized version of AR and TE designs.

8.Conclusion and future scope:

There are various papers discussing hummingbird cryptographic algorithms on different platforms like microcontrollers based on ASIC, spartan-2 FPGA, spartan-3 FPGA, etc. In all of these, there is enhanced research on reducing area, power requirement, & increasing speed with aim of giving better security to resource-

constrained devices like RFID, sensor nodes. In this paper, the proposed hybrid design attempted to enhance the performance and reduce the area consumption of hummingbird algorithms by considering compromise between area reduced and throughput enhanced designs of a hummingbird. It shows that there is a need for further work to enhance the above parameter of hummingbird cryptography.

In this paper, the encryption process of TE, AR, and hybrid designs is implemented. The decryption block cipher, inverse s-box required in decryption process, andimplementation of decryption process of TE, AR, and hybrid designs are also done.

The robustness of the hummingbird cryptographic algorithm is not yet been considered even though there is more scope on checking the robustness of the security algorithms by applying the attacks like the one developed in**(Xinxin Fan and Guang Gong, Honggang Hu-Remedying the Hummingbird Cryptographic Algorithm International Joint Conference of IEEE on Trust, Security and Privacy in computing and communications(TrustCom), Changsha, IEEE, pp772-778, 2011)**. So in continuous with this work, the robustness measurement of hummingbird cryptography can be considered for future work.

Further work is needed to increase the performance and reduce the area and power. It is also possible to design block cipher by rolling only two rounds of operations so it may be considered for further work.

9.References (APA)

A.Poschmann, G. Leander, K. Schramm, and C. Paar, ―New light-weight crypto algorithms for RFID,‖ IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, LA, pp 1843–1846, May 2007.

Axel York Poschmann―LIGHTWEIGHT CRYPTOGRAPHY Cryptographic Engineering for a Pervasive World for the degree Doktor-Ingenieur , Bochum, Germany, February 2009.

Behrouz A. forouzan and debdeep mukhopadhyay ― cryptography and network security, Mc Graw Hill, 2nd edition, 2012.

Biao Min, R. C. C. Cheung and Yan Han, "FPGA-based high-throughput and area-efficient architectures of the Hummingbird cryptography," IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society, 2011, pp. 3998-4002, doi: 10.1109/IECON.2011.6119963.

Biao Min, Ray C.C.Cheung, Yan Han ― FPGA based high throughput and area-efficient architectures of the hummingbird cryptography‖,37th annual conference on IEEE industrial electronics society, Melbourne, IEEE, pp 3998-4002, 2011.

D. Hong et al, HIGHT: A new block cipher suitable for the low-resource device,‖ in CHES 2006, ser. LNCS, L. goubin, and M. Matsui, Eds., vol. 4249. Springer, pp. 46-59, 2006.

Daniel Engels, Xinxin Fan, Guang Gong, Honggang Hu, and Eric M. Smith Hummingbird: Ultra-Lightweight Cryptography for Resource-Constrained Devices, R.Sion et al. (Eds.): FC 2010 Workshops, LNCS 6054, pp. 3–18, 2010. c IFCA/SpringerVerlag Berlin Heidelberg 2010.

De Canniere, C., Preneel, B.: TRIVIUM — Specifications. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/030 (2005).

E Biham, RJ Anderson, LR Knudsen, ―Serpent: A New Block Cipher Proposal‖, in Fast Software Encryption — FSE 98, Springer LNCS vol 1372 pp 222–238.

Emiliano De Cristofaro, Claudio Soriente, Gene Tsudik, and Andrew Williams― hummingbird: privacy at the time of twitter‖ IEEE Symposium on Security and Privacy(SP), San Francisco, CA, IEEE, pp 285-299, 2012.

F. Chabaud and S. Vaudenay, "Links Between Differential and Linear Cryptanalysis", Advances in Cryptology - EUROCRYPT ‘94 (Lecture Notes in Computer Science no. 950), Springer-Verlag, pp. 356-365, 1995.

F.-X. Standaert, G. Piret, N. Gershenfeld, and J.-J. Quisquater, ―SEA: A scalable encryption algorithm for small embedded applications,‖ in Smart Card Research and Applications (Lecture Notes in Computer Science), vol. 3928, J. Domingo-Ferrer, J.Posegga, and D. Schreckling, Eds. Berlin, Germany: Springer-Verlag, pp222–236, 2006.

Gaurav Bansod, Nishchal Raval, and Narayan Pisharoty ‖implementation of a new lightweight encryption design for embedded security ― IEEE Transactions on information forensics and security, IEEE, volume 10, pp 142-151, January 2015.

J. Guo, T. Peyrin, A. Poschmann, and M.J.B. Robshaw. ―The LED Block Cipher‖ In B. Preneel and T. Takagi, editors, Cryptographic Hardware and Embedded Systems - CHES 2011, volume 6917 of LNCS, pages 326–341. Springer, 2011.

Jun Peng, Liang Lei, Qi Han, and Rong Jia ―A chaos-based block cipher with Feistel structure‖, IEEE 13th international conference on cognitive informatics and cognitive computing, London, IEEE, pp 343-348, 2014.

L.Young-I1, L. Je-Hoon, Y. Younggap, and C. Kyoung-Rok, ―Implementation of HIGHT cryptic circuit for RFID tag,‖ IEICE Electronics Express, vol.6, no. 4, pp180-186, 2009.

Lim, C. H, Korkishko T. ―mCrypton - A lightweight block cipher for the security of low-cost RFID tags and Sensors In Information Security Applications. Springer Berlin Heidelberg, pp 243-258, 2006.

Meng-Qin Xiao, Xiang Shen, Yu-Qing Yang, Jun-Yu Wang ―Low Power Implementation of Hummingbird Cryptographic Algorithm for RFID tag 10th IEEE international conference on solid-state and integrated circuit technology (ICSICT), Shanghai, IEEE, pp 581-583,2010.

P.V.G. Raj Pritha, N.Suresh ―Implementation of Hummingbird 1s Cryptographic Algorithm for Low-Cost RFID Tags using LabVIEW ‖ international conference on information, communication and embedded systems (ICICLES), Chennai, IEEE, pp 1-4,2014.

R. RajaRaja and D Pavithra ‖ implementation of hardware efficient lightweight encryption method‖ IEEE International conference on communication and signal processing, Melmaruvathur, IEEE, pp 191-195, 2013.

S.Saha, M. R. Islam, H. Rahman, M. Hassan and A. B. M. A. Hossain, "Design and implementation of block cipher in hummingbird algorithm over FPGA," Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2014, pp. 1-5, doi: 10.1109/ICCCNT.2014.6963084.

Sai Seshabhattar, Priyanka Yenigalla, Paul Krier and Daniel Engels ―Hummingbird Key Establishment Protocol For Low- Power ZigBee‖ The 8th Annual IEEE Consumer Communications and Networking Conference (CCNC)- Security and Content Protection, Las Vegas, NV, IEEE, pp 447-451, 2011.

Sushma Verma, Saibal Kumar Pal, and S. K. Muttoo ―A new tool for lightweight encryption on android‖ IEEE international conference on advanced computing, Gurgaon, IEEE, pp 306-311, 2014.

T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, ―The 128-bit block cipher CLEFIA,‖ in Fast Software Encryption (Lecture Notes in Computer Science), vol. 4593, A. Biryukov, Ed. Berlin, Germany: Springer-Verlag, pp. 181–195, 2007.

T.Eisenbarth and S. Kumar, ―A survey of lightweight-cryptography implementations,‖ IEEE Design and Test of Computers., vol. 24, no. 6,pp. 522–533, Nov./Dec. 2007.

William Stallings,‖ cryptography and network security, principles and practice, Pearson education, 2nd edition, 2002.

Woo Kwon Koo, Hwaseong Lee, Yong Ho Kim, Dong Hoon Lee‖ implementation and analysis of new lightweight cryptographic algorithm suitable for wireless sensor networks‖, IEEE International conference on information security and assurance, Busan, IEEE, pp 73-76, 2008.

Xinxin Fan and Guang Gong, Honggang Hu ―Remedying the Hummingbird Cryptographic Algorithm International Joint Conference of IEEE on Trust, Security and Privacy in computing and communications(TrustCom), Changsha, IEEE, pp772-778, 2011.

Xinxin Fan and Guang Gong, Ken Lauffenburger, Troy Hicks ―FPGA Implementations of the Hummingbird Cryptographic Algorithm ‖ IEEE International Symposium on Hardware- Oriented Security and Trust (HOST), Anaheim, CA. IEEE, pp 48-51, 2010.

Xinxin Fan, Guang Gong, Daniel W. Engels and Eric M. Smith― A Lightweight Privacy-Preserving MutualAuthentication Protocol for RFID Systems ‖ GLOBECOM Workshops(GC workshops) of SCPA and SaCoNAS, Houston, TX, IEEE, pp 10831087, 2011.

Xinxin Fan1, Honggang Hu1, Guang Gong1, Eric M. Smith2, and Daniel Engels― Lightweight Implementation of Hummingbird Cryptographic Algorithm on 4-Bit Microcontrollers international conference for internet technology and secured transaction, London, IEEE, pp 1-7, 2009.

Xunjun Chen, Yuelong Zhu, Zheng Gong, Yiyuan Luo ―Cryptanalysis of the Lightweight Block Cipher Hummingbird-1‖ Fourth International Conference on Emerging Intelligent Data and Web Technologies, Xi'an, IEEE,pp515-518,2013.

Yaser Esmaeili Salehani and Amr Youssef ― differential fault analysis of hummingbird ‖ Proceeding of the international conference on security and cryptography(SECRYPT), Seville, Spain, IEEE, pp357-361, 2011.

Zheng Gong, Svetla Nikova, and Yee Wei Law ―KLEIN: A New Family of Lightweight Block Ciphers In RFIDSec. (2011).