A NOVEL CRYPTOGRAPHIC APPROACH FOR SCADA SYSTEMS USING AES ALGORITHM WITH 256 BIT KEY IN FPGA

By

AMRIK SINGH (SAP Id: 500049028)

(SCHOOL OF COMPUTER SCIENCE)

Submitted

IN PARTIAL FULFILLMENT OF THE REQUIREMENT OF THE DEGREE OF DOCTOR OF PHILOSOPHY

TO

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

DEHRADUN

October, 2019

Under the Guidance of

Dr. Ajay Prasad

Associate Professor, School of Computer Science, U.P.E.S., Dehradun

Dr. Yoginder Talwar

Scientist (D); NIC, NEW DELHI

ACKNOWLEDGEMENT

I convey my sincere thanks to Prof. S. J. Chopra, Chancellor, UPES, Dehradun, Dr. Deependra Kumar Jha, Vice Chancellor, UPES, Dehradun, Dr. Kamal Bansal, Dean, COES and Dr. Manish Prateek for giving me permission for my registration as a Ph.D scholar and for providing vital support for carrying out my Ph.D. research work at School of Computer Science, University of Petroleum & Energy Studies, Dehradun.

I would first like to acknowledge with a deep sense of gratitude for the guidance and advice in research methodologies and support extended by my Ph.D. Guides Prof. Dr. Ajay Prasad, Internal Guide, and Dr. Yoginder Talwar External Guide. Their suggestions, valuable constructive criticism helped me to identify my topic of research in right perspective. I always found my guides spent a long time for technical discussions and to indicate further areas of research on the topic.

I have a deep sense of gratitude for the support and guidance extended by Dr. Rominder Kaur Randhawa, Director, Guru Tegh Bahadur Institute of Technology, New Delhi who has always encouraged and helped me to complete my research work. The thesis would not have been completed without the frequent help of my colleagues at GTBIT, New Delhi. In particular, Dr. Vaneet Singh, Mr. Gurmeet Singh, Mr. Pawan Kumar, and Mr. Gagandeep Singh for the various ways they helped me during my research work.

I am very much thankful to AICTE for funding me Rs. 12,00,000/- (Rs. Twelve Lakhs) under Project Modernization and Removal of Obsolescence (MODROBS) in the financial year 2010-11, in order to establish “Electronics Engineering Innovative Projects Design Laboratory” at Guru Tegh Bahadur Institute of Technology, Rajouri Garden, New Delhi. The VLSI (Integrated Chip) Design Software System of Mentor Graphics EDA Tools, Xilinx VLSI Software ISE System Edition 3.1i, Atlys Spartan-3, Spartan-6 and XUP Virtex-5 FPGA Development kits were very useful in carry out experiments and VHDL software development and testing.

At the end, I thank the members of my family, without whose patience, encouragement and support this work would not have been completed. Last but not the least I thank Ms. Jasvinder Kaur my wife, for the loving support that she provided during this period along with my son, and daughters, son in laws, daughter in law who kept me encouraged to complete my work.

Amrik Singh

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Date:

(Amrik Singh)

CERTIFICATE

This is to certify that the thesis entitled "A Novel Cryptographic Approach for SCADA systems using AES Algorithm with 256 bit key in FPGA", which is being submitted by Mr. Amrik Singh to the Department of Information Technology, University of Petroleum and Energy Studies, Dehradun, for the award of the degree of Doctor of Philosophy, is a record of bonafide research work, he has carried out under our supervision and guidance, and in our opinion, it has reached the standard fulfilling the requirements of the regulations relating to the degree. The results contained in this thesis have not been submitted to any other university or institute for the award of a degree of a diploma.

(Internal Guide)
Dr. Ajay Prasad, Professor
Department of Information and Technology,
University of Petroleum and Energy Studies,
Bidoli, Dehradun.

(External Guide)
Dr. Yoginder Talwar, Scientist –'D'
National Informatics Centre,
Lodhi Road,
New Delhi-110003.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

There is a dire need to protect the Petroleum and other Industries by using the appropriate data security system, cyber attacks from terrorist, national enemies, disgruntled employees are expected now days on an Oil Refineries, On Shore Petroleum fields, Off- Shore Platforms, Oil and Gas Pipe Lines, which will have drastic impact on oil production and in turn reduce economy of the country, it may impact biological damage to the environment. The SCADA systems are used to gather information from field sensor devices, and present a human operator with alarms, current status of the process, performance data, and statistics of real-time processes. The control systems have to respond quickly to compensate for changes within process time-constraints. The SCADA system issue commands in the event of a failure in a process that must meet stringent time constraints, timeliness of message delivery is critical.

Advance Encryption Standard (AES) algorithm implementation in a FPGA has been selected because FPGA scheme has low development cost and requires less development time. The flexibility in design variations is available if required in implementation stage; the security may be moderate to high. The developmental time is low and marketing time is also short. Corporate managers that are located outside the plant needs updating production data from plant manager, but the data need to be highly secured with powerful encryption algorithms with longer security keys, appropriate firewalls, and better protocols for data exchange within the network of SCADA system. It was proposed to have a highly secured AES algorithm implementation in FPGA with 256 bit key size, the design should be a cost effective, minimum development time, with strong security systems. The systems must be tested, analyzed for security performance with respect to the existing available systems. These objectives can be met by the development of an efficient hardware Implementation of AES Algorithm in FPGA with 128 bit key and Highly secure hardware Implementation of AES algorithm in FPGA with 256 bit key size, for data communication between Corporate Business systems and SCADA process control systems of the Petroleum and other Industries.

The hardware implementation design has been coded using VHDL hardware language and all the synthesizing of hardware was done using Xilinx ISE Software 12.4 version and target FPGA device used was xc5vtx240t-2-ff1759, synthesis reports for 256 bit security key and 128 bit security were generated. Simulation results shows that input plain text data is properly ciphered in encryption operation and when ciphered text is given as input to decryption operation, deciphered data is found to be the original input data of encryption operation, the data was tested and analyzed. The system design data was compared with the results reported by other authors. The comparative table clearly shows that our pipe lined architecture using look up tables for S-blocks are better in terms of latency, throughput and higher security with 256 bits security key.

LIST OF PUBLICATION

1. Amrik Singh, Ajay Prasad, Yoginder Talwar, “SCADA Security Issues and FPGA Implementation of AES - A Review”, in IEEE International Conference on Next Generation Computing Technologies NGCT -2016, held at U.P.E.S., Dehradun, on 15th. October 2016, ISBN: 9781509032570.

2. Amrik Singh, Yoginder Talwar, Ajay Prasad, “Highly Secure and Fast AES Algorithm Implementation on FPGA with 256 bit key size”, in International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075 (on line) in Volume -6, Issue No.-7, page -8, December 2016.

3. Amrik Singh, Ajay Prasad, Yoginder Talwar, “Compact and Secure S-Box Implementation of AES - A Review”, in 2nd. International Conference on Smart IOT Systems: Innovations in Computing (SSIC) – 2019, held at Manipal University, Jaipur. Paper sent for the publication in Smart Systems and IOT: Innovations in Computing – Springer book, ISSN 978-981-13-8405-9, Dec. 16, 2-19. https://www.springer.com

CHAPTER – 1

INTRODUCTION

CHAPTER – 1 INTRODUCTION

PROBLEM STATEMENT

Cyber- attacks from terrorist, enemies, disgruntled employees are expected nowadays on an Oil Refineries, Onshore Petroleum fields, Offshore Platforms, Oil and Gas Pipelines, which caters for refined oil production and in turn economy of the country, must be protected, by using appropriate data security system. Since similar arrangements have been made by Australia, as indicated by Christopher (Beggs Beggs, C. 2008), Canada, America and other countries.

The SCADA control applications use either Ethernet or Intranet, but some protocol using TCP/IP provides security with firewalls, cryptography, authentication, intrusion detection systems (IDS) and virtual private network (VPN), suggested by Kim Hyunglun (Wiberg, K. C. 2006) and by Himanshu Khurana (Kaur, A., Bhardwaj, P., & Kumar, N. 2013). In IPsec based VPN each IP packet is encrypted and enclosed within additional IP packet at tunnel ingress and at output side tunnel egress, original frame is extracted and decrypted is suggested (Lamberger, M., et. al 2009). Encapsulating Security Payload (ESP) protocol encrypts the IP packet and Authentication Header ensures security of IP packet, are mandatory parts of IPV6. IPsec devices do internet Key Exchange (IKE) for authentication of peers and distribution of symmetric encryption keys after due negotiation and verification.

All wireless communication may use IEEE 802.11i protocol for high encryption security and use IEEE 802.1x protocol for authentication by verifying user certificates. Data pipe line for all rounds of AES algorithm hardware implementation and using look up table for S-boxes of AES to reduce encryption latency. Trusted database of authorized and legitimate users must be created for verification of dialers, while reading and writing data through modems by means of callback system. Authorized dealer should be allotted unique usernames and passwords. Remote control software should have generation of audit logs, and use encryption for communication of data. The

Challenge Handshake Authentication Protocol must be used for authentication in Link layer.

BACKGROUND

Cyber - attacks from terrorist, enemies, disgruntled employees are expected nowadays on an Oil Refineries, Onshore Petroleum fields, Offshore Platforms, Oil and Gas Pipelines, which caters for refined oil production and in turn economy of the country, must be protected, by using appropriate data security system. The automated control system sends command signal to affect corrective action within process time constraint as per recommended system design. The local disturbance in the utility, in particular short circuit protection must be carried out in 4-40 milliseconds time. SCADA system calls for proper time response for corrective action, reliability, plant safety, personal safety, product quality without any hindrance because of security mechanism. Increased business competition demands higher management efficiencies and protection of sensitive data, which is provided by IT network system by means of confidentiality, authentication, integrity and no repudiation as suggested by NIST Guide for ICS (Matsui, M. 1993), and by Hugh Njemanze (NIST Special Publication 2011). The systems with thousands of monitored nodes with real time responses meet emergency and fluctuation responses.

The remote monitoring of flow rates and pressures in a process are done by SCADA system, based on received new data, industrial control system generate control commands to the appropriate valves and switches. At central control Centre a computer program or an operator generate data for command to balance the flow of material to activate valves and regulators, for corrective action. The vulnerabilities exist in old legacy SCADA systems because Intrusion Detection Devices and advanced encryption algorithm are not implemented in them for protection. (Coates, G. M., Hopkinson, K. M., Graham, S. R., & Kurkowski, S. H. 2008, 2009) and (Mentens, N., Batina, L., Preneel, B., & Verbauwhede, I. 2005). The monitoring of Oil and Gas pipelines infrastructure, offshore drilling, refineries, methane leakage around natural gas extraction can be efficiently done by SCADA software using wireless

communication that to at less cost, as compared to Satellite communication. SCADA software prepares schedule for monitoring data using burst communications any time, day, night.

MOTIVATION AND NEED OF DATA SECURITY USING ENCRYPTION TECHNOLOGIES FOR RESEARCH

A cyber- attack by a terrorist SCADA system, DCS system of an Oil Refinery, Petroleum Infrastructure will have catastrophic impact on petroleum products, a protection strategy for designing an appropriate data security system must be initiated. Although control systems send, command through either Intranet or Ethernet but now some of operations use TCP/IP protocol for data transmission between Local Control Rooms (LCR) and Central Control Room. Local server of each DCS is connected to system server.

AUTHENTICATION MECHANISM OF THE SYSTEM

The CCR server system needs to maintain public keys of all the LCR systems and LCR system should know the public key of the CCR server system. During installation of the LCR system, a public and private key pair is generated, using Digital Signature algorithm software. System administrator giving LCR system ID, public key and IP address to database transfers public key to CCR server system manager. The private key is encrypted and then only stored.

The Authentication protocol mechanism allows LCR system and CCR server to authenticate with each other and negotiate on symmetric cryptographic key of Advance Encryption Standard (AES) before transmitting any field or an application data to CCR server system. The mechanism provides entity ID, key authentication, key confirmation, and key freshness guarantees for the agreed session key. Every time the field data or application is too communicated from LCR to CCR server or command data from CCR server to LCR system to be communicated, a new session key is to be allocated.

Whenever a large amount of encrypted data is transmitted, the attacker may accumulate large encrypted data and attempt to crack the session key used for communication. The session key should be changed if it lasts for more than one hour or if more than 500MB of data has been exchanged as pointed by Kim Hyunglun (Wiberg, K. C. 2006). The CCR server keeps track of the above parameters for each LCR system and initiate key reset after the end of current session key.

SYMMETRIC KEY ENCRYPTION ALGORITHM TECHNOLOGY IMPLEMENTATION

Efficient FPGA Implementation with 128-bit key will be useful for portable security system; to be used by field engineers and third party Contractor hired to run wells, platforms and pipelines, for onshore fields and offshore platforms to enter data in the system. Encryption security key length of 128 bit is sufficient for portable field data entry system. Corporate managers require access to updated data from plant manager of petroleum process, but the data need to be highly secure with powerful encryption algorithms with longer security keys, during data transmission, and better protocols for data exchange within the network of SCADA system. It is proposed to have a highly secure AES algorithm implementation on FPGA with 256-bit key size. Immediately the individual round key is generated, it can be used for processing the data for that specific round, rather than waiting for generation of all round keys, this is the novelty of this technique of round generation.

REVIEW OF LITERATURE

The hardware implementation of AES for smart cards is optimized for area. The FPGA scheme may be implemented in small area for mobiles systems and home applications, which also prefer low power consumption. (Canright, D. 2005 and Canright, D., & Batina, L. 2008), and by Satoh et al. (Rais, M. H., & Al Mijalli, M. H. 2012). The cost of implementation varies as per level of security desired. In physical cyber- attacks the exploitation of the “side channel information”, typically time, power consumption, electromagnetic emission, which can be measured while the cryptographic algorithm is being computed on the device.

Without knowing the technical specification of the chip of mobile system, power analysis attack

can be carried out as pointed by (Marc Joye et al. Intel 2012), to counter this, one has to remove the correlation between the secret key and power consumed and modifying the power characteristics of the device. The Boolean masking technique proposed by (Y. Ishai, Intel® Advanced Encryption Standard New Instructions (AES-NI) 2012), is used to randomize the power consumption by adding a random number, called mask, to all the intermediate values which may be exploited by the attacker. To ensure the correctness of the results, the mask is removed at the end of computation. Masking is implemented to enhance the security level against the power analysis attacks but it incurs significant overhead when applied to S-Boxes of AES, which is a nonlinear transformation. Researchers have proposed various compact designs for masked AES S- box against second-order differential power analysis, (Ozturk, M., & Aubin, P. 2011).

In smart card applications, basic optimized area architecture of AES implementation for one round encryption / decryption in one clock cycle is designed, which is reused ten times for 10 clocks from the data entrance and then encrypted / decrypted data will be available at output.

In network routers applications, a high throughput optimization based on pipelined architecture is designed for high speed operation; by implementing hardware of Subbyte transformation in terms of Galois Field (28) of composite field arithmetic (CFA) calculated circuitry and applying deeper level of pipelining to improve the throughput. In LUT approach the delay due to the time required to pass through FPGA block memories is high. Centre for Protection of National Infrastructure (CPNI) (Centre for the Protection of National Infrastructure (CPNI), USA, 2006) has suggested the good practices to be observed for protection of SCADA systems from cyber - attacks.

Terrorists and Hackers exploit TCP / IP Network and the Operating System vulnerabilities. Rarely actual PLC or RTU are exploited. We must carefully test and evaluate the software tools before deploying them in the Infrastructure. The common wisdom is that never mix office LAN with SCADA System and should be separated by proper Firewall, or at least a bridge or a router. Getting sensor operational data from fields, processing, displaying data information, and relaying control commands to remote or local SCADA equipment has to be protected being critical Infrastructure. North American Electric Reliability Corporation (NERC) has framed critical infrastructure protection standards known as NERC- CIP, which has been supported by US Energy Policy Act of 2005.

The under mentioned standards for SCADA communications to provide security through Encryption and Authentication were developed.

IEEE 189 Suite: For SCADA communication.

IEC 62351 Suite: For Secure Authentication of DNP3 Communication.

NIST, USA has published Guidelines for SCADA Systems, and Control Systems configurations such as PLC.

DATA ACCESS BASED ON OPERATOR TASK REQUIREMENT BASIS

The task requirement of an operator may be to access a few data points to view status of a machine, process and then control it, although SCADA server provide the control of browsing, reading and writing for an operator. However, the control system Engineer has full read and write access to all the points of the process for full automation. The control Engineer allocates the appropriate data access facility to all operators for optimal operation of the system and to prevent accident by un-authorized access.

The corporate company expects the following improvements by implementation of new automated SCADA system.

a) Reduce outage minutes with restoration of major load.
b) Reduce operations and maintenance costs.
c) Improve coordination with plant substation.
d) Improve operational efficiency.
e) Reduce outages with auto-sectionalizing.
f) Improve load balancing.

In any IT systems ensures Integrity, authentication, availability, non-repudiation and confidentiality. Corporate IT Network performs backups, software up-gradation as per schedule regularly for management system efficiency improvement, for which routinely scheduled downtime is allowed in IT Organizations. However, the downtimes cannot be tolerated in any process SCADA Systems. SCADA system demands plant safety, personal safety, good product quality, real time corrective response, tolerance of emergencies and ensure reliability.

The SCADA control engineer allows access to process operators only after automatic their verification by means of authentication and digital certificates.

SCADA SECURITY ISSUES

a) IEDs, PLCs, RTUs, are selected based on their efficiency ruggedness and real time constraints to prioritize task execution using microprocessors with limited memory and computational capacity, and low bandwidth links.

b) Encryption must be implemented by using cryptographic keys, digital certificates, and digital signatures. SCADA test beds to evaluate the most effective security have been developed by various national laboratories; organizations such as the National Institute of Standards and Technology (NIST) have invested a lot on SCADA security.

OBJECTIVES OF THE RESEARCH PROBLEM

It is observed from the literature that a strong algebraically encryption algorithm with sufficiently long security key be implemented using FPGA hardware implementation with low latency. Therefore, it is proposed to design a cost effective, development time effective, with strong security systems. The systems must be tested, analyzed for security performance with respect to the existing available systems. These objectives can be met by the development of the under mentioned Advance Encryption Standard (AES) systems, for data communication of SCADA systems of the Petroleum Industry.

The objectives for this research are:

a) **Efficient Implementation of AES Algorithm in FPGA with 128 bit key**
b) **Highly Secure AES Algorithm Implementation in FPGA with 256 bit key**

RATIONALE OF THE STUDY

To counter cyber - attacks, early intrusion detection systems should be installed, using correlation techniques to observe abnormal network traffic behavior. On analyzing abnormal traffic events flowing in the networks, corrective measure must be implemented. Powerful encryption algorithms with longer security keys, appropriate Firewalls, better protocols for data exchange within the network should be used to make SCADA system highly secure. A firewall must be deployed to filter out unwanted interference coming from Business network to SCADA control network. Demilitarized Zone routing policy, access lists generation and implementation must be used to safeguard SCADA control systems.

RESEARCH METHODOLOGY ADOPTED AND IMPLEMENTATION

Block Cipher: A symmetric key block cipher takes input of 128-bit group, process a defined sequence of its four transformations, outputs corresponding

ciphered data of 128 bit. The security level of processed data is defined by the size of selected key of 128 bit, 192 bit or 256 bit, to provide different level of security of encrypted data. In decryption process, the ciphered data is taken as input to produce the original 128-bit plain text, when inverse transformations are applied during decryption in reverse order. Messages of large size are converted into data of 128-bit size by using operation mode of Electronic Code Book, Output feedback mode, cipher feedback mode and Cipher block chaining mode.

In AES encryption, the input plain text and output cipher text with a block size of 128 bits and can be viewed as a 4x4 matrix of 16 bytes arranged in a column major format. It can use a key size of 128, 192, or 256 bits and correspondingly has 10, 12 or 14 iterations of round transformations respectively. Each round transformation has four sub transformations namely; Byte Substitution (BS), Row Shift (RS), Mix Column (MC), and Add Round Key (AK). In the last round Mix Column (MC) transformation is not included. Byte Substitution is a nonlinear transformation, which create confusion in the encrypted data. We can make partial linearization in Byte Substitution. (Kim, H., Hong, S., & Lim, J. 2011).

The key expansion mechanism is used to derive round keys from user defined cipher key as per key schedule. The total number of expanded key bytes required for a complete cipher run is equal to the no. of block length bytes (N_b) multiplied by the number of rounds (N_r) plus one. i. e. $N_b(N_r+1)$. Thus the total number of expanded key bytes for key size of 128,192, and 256 bits is

going to be (16X11=) 176, (16x13=) 208 and (16x15=) 240 bytes respectively. The key expansion mechanism for 256 bits key size is considered to be the most secure for data block size of 128 bits whose implementation using FPGA will be discussed in this paper. Use of 256 bits key size in hardware encryption, for highly secure applications such as SCADA systems for Gas, and Oil Pipelines, also Oil refinery is very appropriate.

The AES algorithm may be implemented by following schemes for secured data communication.

a) **Software Schemes**: Software programs schemes are easier to implement, low in cost. They offer a limited physical security and the slowest process. It is likely to be corrupted due to viruses. Due to growing requirements for high speed, high volume secure communication combined with physical security, hardware implementation of cryptography takes place.
b) **VLSI /ASICS Schemes**: This ASICS design schemes have very high development costs and require long development time, however the cost per chip may be low if quantity produced is very high, which is normally low for security devices. The flexibility in design variation is not available. Researchers (Mangard, S., Pramstaller, N., & Oswald, E. 2005), have tried an implementation of S-Boxes of AES in the past to make compact cryptosystem. The optimum design for AES S-Boxes has been attempted by researchers M.M. Wong, et al. proposed construction of Optimum CFA for Compact High-Throughput AES S-Boxes, using CFA with isomorphic mapping, which results in the reduced implementation chip area. The optimization of CFA combinatorial circuit in the field of mapping, basis representation, selecting appropriate field polynomials and isomorphic mapping helps to identify a short critical path for VLSI architecture implementation.

c) **FPGA Schemes**: FPGA implementation schemes have low development cost and requires less development time. The flexibility in design variations is available if required in implementation stage; the security may be moderate to high. The developmental time is low and marketing time is short.

The research proposal has used an FPGA implementation of AES encryption/decryption with data block size of 128 bits and key size of 256 bits, simulation, synthesis reports have been generated, and the results have been compared with the implementations done in the past by other researchers. Our

research proposal has key expansion module to generate round keys calculated as per the general guidelines indicated by National Institute of Standards and Technologies (NIST) documents of USA. Our proposal has used lookup table approach implementation for S-box to achieve low latency and high throughput.

Two designs of FPGA implementations of 128 bit data block size with 128 bits security key and 256 bits security key respectively have been completed. The design has been coded using VHDL hardware language and all the results are synthesized based on Xilinx ISE Software 12.4 version and target device used was xc5vtx240t-2-ff1759. We find encrypted data at transmitter output as quite in random order, since AES algorithm ensures good dispersion and confusion of transmitted data.

S-Box of AES algorithm is implemented normally by using look up tables (LUT) in which 256 predefined values of S-Box and the same numbers for Inverse S-Box are stored in a ROM, it offers a shorter critical depth, it is suitable for FPGA implementation in terms of gate count. In high speed pipelined designs unbreakable delay of LUT becomes drawback. The efficiency of AES hardware implementation in terms of speed, security, size and power consumption largely depend on its architecture Every attempt have been made by researchers to optimize one or more parameters for some specific application, either to reduce the chip area, power consumption or to increase efficiency, throughput, and security level. The different applications of society requirements demand different parameters with respect to size for mobile applications, high speed processing for quick response, (Sasaki, Y. 2011), by researcher (Uddin, M., & Rahman, A. A. 2010). . Architecture in VLSI was proposed for single FPGA chip pipelined design by Kenneth Stevens et al. (Schramm, K., & Paar, C. 2006), for high throughput with fully pipelined FPGA implementation (Kim, J., Hong, S., Sung, J., Lee, S., Lim, J., & Sung, S, 2003) also proposed architecture. Design can be based on logic synthesis using Truth table **or** direct implementation of Algebraic Normal Form (ANF) expression for each column.

S-Box transformation in AES Implementation is the non-linear transformation and it provides confusion part in encryption of data processing and contributes significant part in achieving high security. CFA based optimization is used for reducing area for FPGA or VLSI designs for compact mobile applications, the data security is ensured by adopting different masking techniques. Algorithmic and CFA architectural optimization can be achieved in basic representations by elimination of redundant common factors in the inverter, appropriate choice of the field polynomials is required, and minimize the arithmetic complexity by merger of some multipliers with some sub-operations. The sum of the upper and lower halves of each factor can be shared between two or more sub-field multipliers which have the same input factor, one XOR addition is saved in 2bit factor shared by two GF (2^2). 5 XOR s are saved in 4bit factor shared by two GF (2^4) multipliers. Area saving is achieved on combining GF 2^2) multiplier with a scalar in a GF (2^4) multiplier, their results a saving of 3 XORs in total gates and one XOR in critical path. On combining the sum of upper and lower halves of the inputs of multiplier, common factors with GF (2^4) and square scalar there will be reduction of 2 XORs inverter. We can save around 30 XORS gates in the total gates and 3XORS gates in the critical depth.

The common and straight forward implementation of the S-Box for SubByte transformation is by using pre-computed values stored in PROM based on Lookup table for encryption of data and the InvSubByte transformation by using another Lookup table of inverse S-Box for decryption to obtain decipher data in the receiver output. The different applications of society requirements demand different parameters with respect to high throughput rate for server application, compact in size for mobile applications, high speed processing for quick response and high security level by long security key size. S-Box transformation in AES Implementation is the non-linear transformation and it provides confusion part in encryption of data processing and contributes significant part in achieving high security.

CRYPTANALYSIS OF AES ALGORITHM

The basic security primitives used for constructing security solutions are messages authentication codes, authenticated encryption algorithm, hash Functions, stream ciphers, pseudo random number generators and entropy extractors, block ciphers primitive is considered as well understood, and was recommended for symmetric key encryption standard by USA, first for DES and for AES.

A block cipher with n bit block and k bit key is a subset of 2k permutations among all 2n factorial permutations on n bits to convert n bit plaintext data block to n bit cipher text data block. After initial addition of data with sub-key, Substitution Box function is used for local non-linear operation and bit permutation and matrix vector multiplication for linear operation in most of ciphers i. e. AES, DES, Present, Camellia, Clefia. Serpent and hash functions i.e. photon, Spongent. Whirlwind, and Groestl.

VARIOUS TYPES OF CYBER ATTACKS

The followings are the different cyber- attack method for cryptanalysis:

a) Brute-force attack
b) Davies' attack
c) Differential Linear cryptanalysis
d) Truncated cryptanalysis
e) Higher-order differential cryptanalysis
f) Boomerang attacks
g) Impossible differential cryptanalysis
h) Improbable differential cryptanalysis
i) Integral cryptanalysis
j) Linear cryptanalysis
k) Multiple – approximation
l) Meet-in-the-middle attack
m) Mod-n cryptanalysis

n) Related-key attack
o) Sandwich attack
p) Slide attack
q) XSL attack

FAST AND SECURE AES IMPLEMENTATION

The processing of AES transformations by conventional processors gets speed limited. The high speed intellectual processor cores (IP) dedicated processors with new long instruction sets have been developed by Intel Corporation, to accelerate the performance of Galois Field fixed field constant multiplication, an important element of AES algorithm, in comparison to pure software implementation speed. An instruction of Intel PCLMULQDQ for Intel Core processor can perform carry-less multiplication of two 64 bit operands, without propagation of carry values, by computing Galois Hash for efficient implementation of AES. A 127- bit output of two 64-bit operands is produced, which in turn may be used by software for generating the 255-bit output for GCM. The most significant bit equals 0 among 256 bit result.

Galois Counter Mode generates the message digest termed as Galois Hash from encrypted data meant for message authentication. The previous Galois Hash value is XOR- ed with the current cipher text block. The output is multiplied in GF (2^{128}) with hash value. Irreducible polynomial.

$$g = g(x) = x^{128} + x^7 + x^2 + x + 1$$

is used to produce GCM.

VPN TECHNOLOGY AND IPSEC STACK OF PROTOCOLS

The researcher (Bollapragada, V., Khalid, M., & Wainner, S. (2005) suggested the use of VPN technology and (Gepner, P., & Kowalik, M. F. 2006) proposed the IPSec Stack of security protocols which he consider as one of the efficient approaches to significantly reduce security concern in computer network for transmission of RMM data Video, audio and graphics, etc. over public Internet.

Computer Security Institute published the report (2010) regarding the user's satisfaction rates with respect to computer network security and observed high satisfaction level for Firewalls provision, Encryption for data transmission, VPN, and one time pass word for Smart cards. The researchers (Trichina, E., & Korkishko, L. 2004), (Bollapragada, V., Khalid, M., & Wainner, S. (2005) suggested the key aspect of mobile (MVPN) design, development and implementation set of security, data exchange protocols, communication and dynamic VPN tunnels for mobility and security.

PROTOCOLS IPSEC STACK OF SECURITY

IPSec is an open standard for providing private secure communications over Internet Protocol (IP) protocols, by identifying various encryption and authentication algorithms to be used and provide cryptographic keys required for services. Security related problems of transfer of confidential data related to rich multimedia data of video, audio and graphics over Internet are ensured by the development, design, and implementation is a data exchange protocol of communication, security set is a key aspect of dynamic VPN tunnels for user security requirements.

The Thesis has the following six Chapters

Chapter No.1:

Problem Statement states the type of cyber-attacks from terrorist and disgruntled Ex. employees and enemies to our petroleum automation and other Industries utilities. The need of using appropriate data security systems for protection of our critical SCADA controlled Industrial Infrastructure. The SCADA control applications use either Ethernet or Intranet, but some protocol using TCP/IP provides security with firewalls, cryptography, authentication, intrusion detection systems (IDS) and virtual private network (VPN),

Chapter No.2:

The Various Architectures of the SCADA System has been explained. The benefits of SCADA Systems and the need to secure SCADA System from cyber-attacks from competitors and ex. disgruntled Employees has been highlighted. The comparison of various open source Intrusion Detection Systems have been explained. Intrusion prevention systems provision must be deployed. The Distribution Network Protocols architecture has been explained with its Analyzers, DNP-3 Parser, its Implementation and its performance evaluation have explained. The modern SCADA systems, Cyber security threats and Strategy, Global SCADA System and Enterprise performance ecosystem have been explained

Chapter No.3:

The various Implementations schemes of AES algorithm for encryption of data at Transmitter and decryption of data at Receiver using Field Programmable Gates Arrays (FPGA) chips have been explained with Diagrams. Top Level Entity Implementation Block Diagram for AES algorithm with security key of 128 Bits is given. Performance Comparison of implementation with earlier Implementers has been given. The simulation results of AES Implementation for input data of all zeros and ones have been generated and shown results. Instruction Sets of AES-NI meant for increasing the speed of AES Implementation and increasing in-build data security explained. Various high

speed and secure AES Implementations, Authenticated Encryption with Associated Data (AEAD) Modes, VPN Technology and IPSec Stack of Protocols, EAX mode of Cipher Operation have been explained to increase speed of implementation and security.

Chapter No.4:

Practical Implementations of Substitute Box (S-Box) of AES algorithm are explained. The Look up Table (LUT) Implementation has an unbreakable delay to pick up value of S-Box from PROM stored chip. Another Combinational Logic Circuits Implementation has the limitation of not very fast in speed and size of circuit not small. The Composite Field Architecture (CFA) Implementation may be small and fast in speed for high throughput applications since it can be optimized for algorithmic and architectural. The multiplicative Inversion technique and by using isomorphic mapping with common sub expression elimination in sub field helps in reducing chip area. FPGA Implementation using CFA technique in achieving high-speed data processing.

Chapter No.5:

The generation of new method for individual round keys from the given security key of 256 bits of AES have been adopted and analyzed for implementation. The Notations and Notions have been proposed and then calculated the every individual round keys from the given security key. After all the round keys are generated, these may be stored until the given is in use. Immediately the individual round key is generated, it can be used for processing the data for that specific round, rather than waiting for generation of all round keys, this is the novelty of this technique of round generation.

Chapter No.6:

The high level of security in the data processing of encryption can be achieved using security key of the size of 256 bits in length. The use of new Instructions set of AES-NI has in built high level of data security. The implementation of AES algorithm for processing data for encryption with security key of 256 bits

has been implemented using FPGA chip no. XC5vtx240t-2-ff1759 with different sets of input data. The simulation results of processing data and generation of in between data processing was generated and found correct values of ciphered data and generation of original data at receiver output. Synthesis Reports of chip design for the FPGA chip no. XC5vtx240t-2-ff1759 has been generated and attached. Comparison Table of our chip design and earlier researcher is attached.

CHAPTER -2

SCADA

CHAPTER -2 SCADA

SCADA SYSTEMS

SCADA process automation system update the process data as received from field sensors, transducers and instruments and sends to control room computer for controlling and monitoring purposes. Operator driven or automated commands can be transmitted to remote field devices. Programmable Logic Controller (PLC) and Remote Terminal Unit (RTU) receives data from sensors, converts signals into digital data and outputs to supervisory system. PLCs are configurable, flexible, and versatile than RTUs.

The Communication Infrastructure, may be wireless, cables, satellite, or combination of these are used for data transfer between field data interface devices and control and supervisory system. Host Computer/ Master Terminal Unit (MTU)/ SCADA Server is used to storing databases, human monitoring and process controlling, displaying statistical control charts and reports. Human - machine interface (HMI) is a device to present process data to the operator.

In legacy SCADA systems, the security concerns are the minimum, because it used proprietary networks. IP based SCADA systems are designed now days for increased efficiency, reduced business management cost, readily available production process data, from corporate server. These systems are distributed, and networked using open protocols of TCP/IP of internet and make them vulnerable to cyber terrorism. In spite of adopting excellent management, practices in managing IP based SCADA as suggested by National Institute of Standards Technologies (NSIT), USA; one has to look into likely security concerns and vulnerabilities, identify proper security management methods to overcome them.

The strict security for the internal network and the systems in demilitarized zones (DMZs) must be enforced, virtual private networks (VPNs) should be used for enhancing security. Thoroughly inspection on regular basis of security and the vulnerability should be evaluated. Concentration of monitoring should high and access paths to internal network should be the minimum. In case of

contingencies, monitoring methods and developing controls of SCADA equipment should be planed for implementation.

SCADA ARCHITECTURE

Internet Protocol is very popular because it can used over any kind of media, wired (Optical fiber, telephone lines, ADSL, Cable) and wireless (spread spectrum, satellite, radio, WLAN, or cellular,). Final decision of the architecture depends on the data rates, installation budgets, existing communication infrastructure, remoteness of the site and available communication at isolated field sites, future needs polling frequencies, and requirements. Distributed SCADA processing functionality at physically separated locations using WAN may be able to partially reduce losses in case of natural environmental disaster. During the designing, implementation and testing of SCADA system, security parameters must be checked and verified by following appropriate designed procedures at the time of installation. Standard ISA S99 security level model must be followed, use equipment that provides operation on two layers only. IP communications from untrustworthy networks to SCADA should terminate at buffer network only. Solid defense must be created using well designed firewalls for blocking suspected packets, and Intrusion Detection System (IDS). Internet Protocol version 6 (IPv6) should be used in designing new SCADA system, since it has auto configuration, extensibility, mobility, and large address space of 128 bit in comparison to 32 bit of IPv4 (Alsiherov, F., & Kim, T. 2010) and IPv6 based network architecture for wireless sensor network has been designed and tested for production quality implementation.

Designs and procedures are crucial components, which must ensure that all security requirements are recognized during design phase, implemented, and tested at the time of installation. Standard ISA S99 security level model must be followed, use equipment that provides operation on two layers only. SCADA communication should be encrypted and routed through a VPN tunnel which runs through corporate IT or through existing non-critical networks. IP communications from untrustworthy networks to SCADA should terminate at

buffer network only. Solid defense must be created using well-designed firewalls for blocking suspected packets, and Intrusion Detection System (IDS).

Public key algorithm to encrypt session keys and symmetric algorithms for encryption and decryption of data for SCADA system must be implemented. Internet Protocol version 6 (IPv6) should be used in designing new SCADA system, since it has auto configuration, extensibility, mobility, and large address space of 128 bit in comparison to 32 bit of IPv4. IPv6 based network architecture for wireless sensor network has been designed and tested for production quality implementation by Hui and Culler. Low Power wireless personal area networks (6LoWPAN) have been designed using packet format standardized by the IETF to enable IPv6 communication over LoWPANs as per RFC 4944 standard.

LIKELY THREATS AND THEIR PREVENTIONS

Spectrum techniques for radio communication should be used to prevent threats of eavesdropping and tampering or jamming radio signal in the physical layer. Admission control mechanism for ignoring excessive request without identifying authenticity should be used to prevent induction of a collision, contention or by deliberating fragmenting packets to bypass the Intrusion Detection System, in Data Link layer. Message modification, fabrication and interruption can be prevented in Network layer by enforcing encryption. The malicious node should be detected, isolated and removed from the network by using authentication and encryption mechanism. Wormholes, Sybil, and Sinkholes attacks against routing protocols can be prevented by employing a durable key management. In transport layer, session hijacking and flooding attacks should be prevented by using proper authentication mechanism or by controlling the number of connections nodes can make. In application layer malicious code attacks can be prevented by its detection and then isolating them.

AUTHENTICATION MECHANISM SYSTEM

The Central Control Room (CCR) Server system needs to maintain public keys of all the Local Control Room (LCR) systems and LCR system should know the public key of the CCR server system. During installation of the LCR system, a public and private key pair is generated, using Digital Signature algorithm software. System administrator giving LCR system ID, public key and IP address to database transfers public key to CCR server system manager. The private key is encrypted and then only stored.

The Authentication protocol mechanism allows LCR system and CCR server to authenticate with each other and negotiate on symmetric cryptographic key of Advance Encryption Standard (AES) before transmitting any field or an application data to CCR server system. The mechanism provides entity ID, key authentication, key confirmation, and key freshness guarantees for the agreed session key. Every time the field data or application is to communicated from LCR to CCR sever or command data from CCR sever to LCR system to be communicated, a new session key is to be allocated.

Whenever a large amount of encrypted data is transmitted, the attacker may accumulate large encrypted data and attempt to crack the session key used for communication. The session key should be changed if it lasts for more than one hour or if more than 500 MB of data has been exchanged. The CCR server keeps track of the above parameters for each LCR system and initiate key reset after the end of current session key.

SYMMETRIC KEY ENCRYPTION ALGORITHM TECHNOLOGY IMPLEMENTATION

Efficient FPGA Implementation of AES algorithm with 128 bit key will be useful for portable security system to be used by field engineers and third party Contractor hired to run wells, platforms and pipelines, for on shore fields and off shore platforms to enter data in the system. Encryption security key length of 128 bit is sufficient for portable field data entry system and remote

monitoring and sending commands to valves and switches, in gas and oil pipelines where they monitor flow rates and pressures.

DEMILITARIZED ZONE (DMZ)

DMZ is a good technique for securing communication network based on the principal of network separation strategy between secured network (SCADA, DCS LAN) and Corporate Intranet, which can be further separated with another DMZ from internet if connected. Double protection is provided by the use of two DMZ, first between SCADA system and Corporate System and second between Corporate Intranet System and Internet. DMZ is a zone between an inner firewall and an outer firewall, ensure that server and database resides in safe place. The firewall acts as a filter that permits the data to enter from selected ports and blocks others. It is properly configured to protect passwords, IP addresses, and files. At DMZ output, a router is used as border router to route information to correct destination.

BENEFITS OF SCADA SYSTEMS

The corporate company expects the following improvements by implementation of new automated SCADA system.

a) Reduce outage minutes with restoration of major load.
b) Reduce operations and maintenance costs.
c) Improve coordination with plant substation.
d) Improve operational efficiency.
e) Reduce outages with auto-sectionalizing.
f) Improve load balancing.

National Laboratories of some countries have initiated programs focusing on SCADA security, by establishing test beds to evaluate the effectiveness of the security of SCADA.

TYPICAL SECURED SCADA SYSTEMS

A simple secured SCADA system will comprise of a Corporate Server supported by Data Historian, Work Station, and a Printer in minimum configuration but protected by a Router and Firewall from a possible attack from Internet connection. Control Server supported by Human Machine Interface (HMI), Engineering Work Station in minimum configuration is protected by another set of Firewall and a Router in between Corporate Business systems, as shown in Figure 2.1. PLC's monitoring pollution sensors, temperature sensors, noise level sensors, water level sensors, flow rate sensors, pressure sensors, oil level sensors etc. and controlling actuators of solenoid valves, pressure regulators, servo drives, variable frequency drives, temperature regulators, humidity regulators etc. are connected to Remote Terminal Units (RTU), which are in torn connected to Control Server through Satellite, Power Line Carrier Comm. (PLCC), WAN Network, Microwave/ Radio Comm., Cellular comm. via Front End Processor (FEP), as shown in Figure 2.2 .

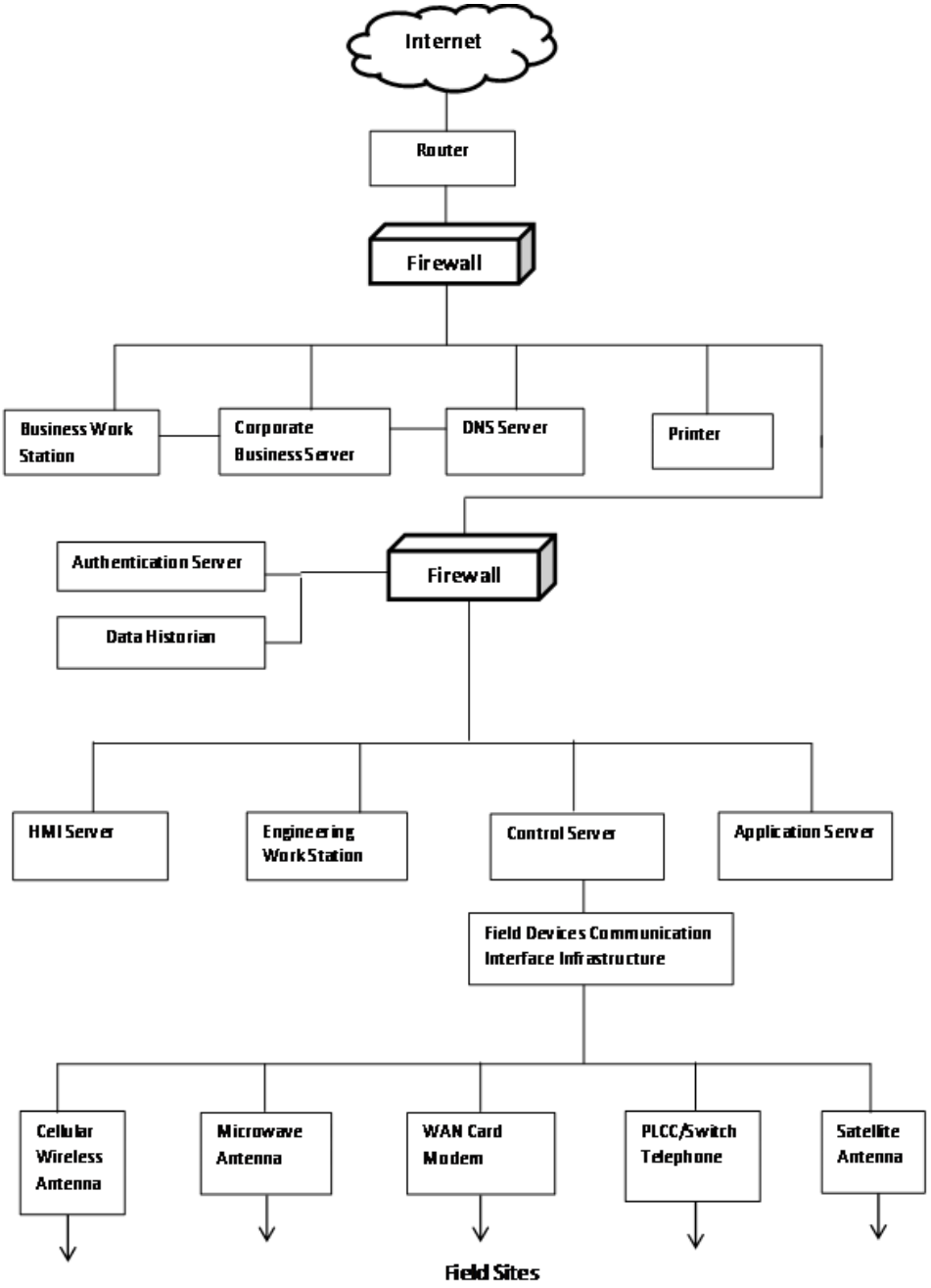

Figure 2.1 Typical Secured SCADA System

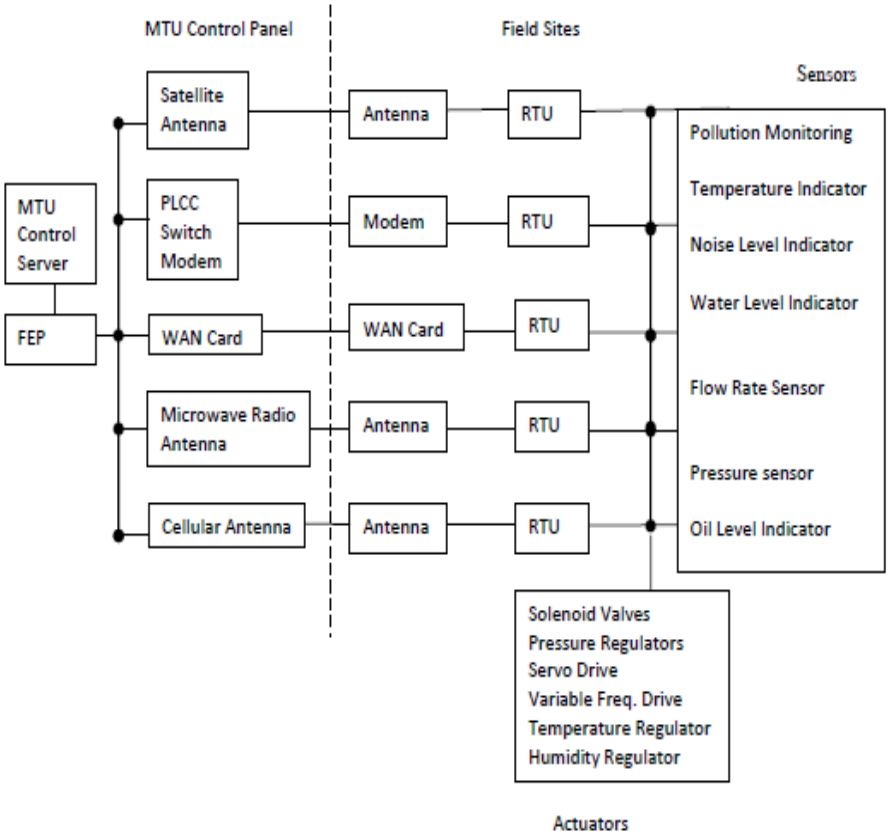

Figure 2.2 Process Control SCADA System

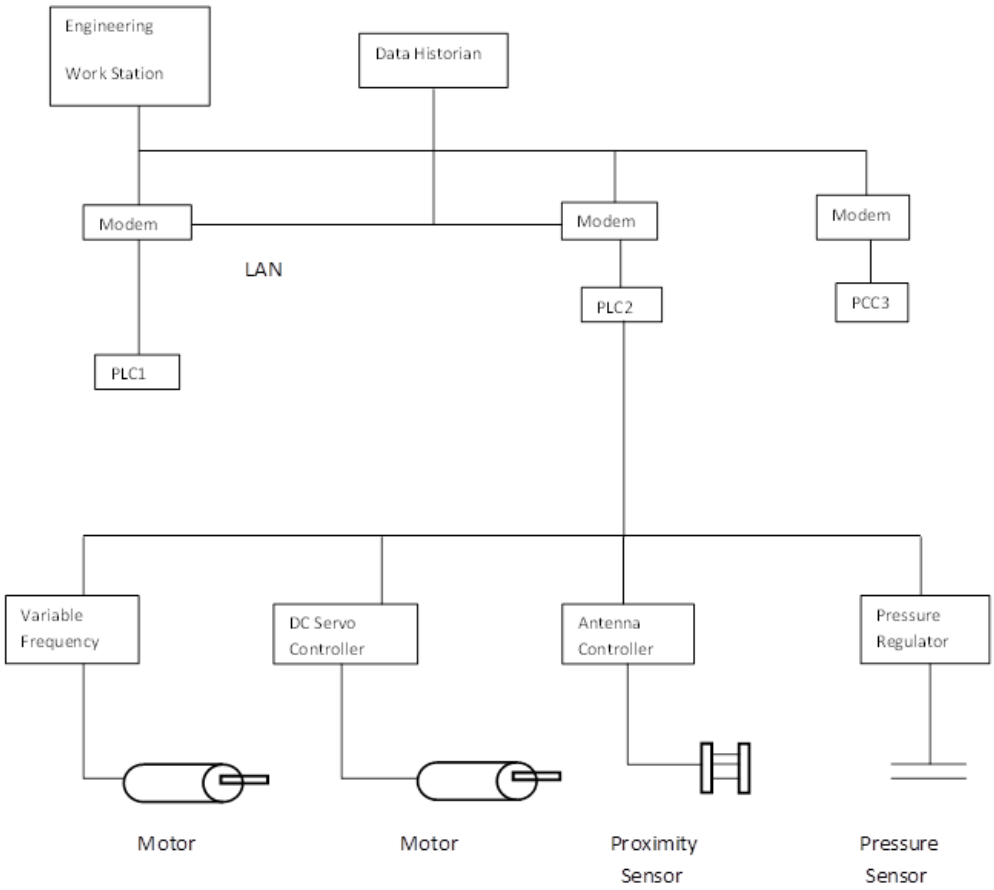

Figure 2.3 PLC Control System Implementation

A LARGE PROCESS SECURE CONTROL SYSTEM

Large Process Control Server may be supported by Data Historian, Data Acquisition Server, Engineering Workstation, Database Server, HMI Computer, and Configuration Server, all connected in Control System LAN configuration. Corporate Business Server is supported by Web Application Server, Workstation, Authentication Server, FTP Server, DNS Server, and email Server; all of them are connected through Corporate Firewall to Internet Infrastructure. (Alsiherov, F., & Kim, T. 2010). .

Control System LAN is protected from cyber-attacks, by using control system Firewall between corporate LAN and control LAN and creating Demilitarized Zone (DMZ) for Authentication Server, Security Server, and Data Base Historian, as shown in Fig. 2.3. All field process devices i.e. Intelligent Electronics Devices (IEDs), PLCs, RTU's and Controllers are connected to

Control Server through Modem Pool and Communication Interface Infrastructure.

Firewalls must be configured with tightly rule bases, should have regular reviews, be managed by trained administrators, changes managed under strict control. Management and monitoring of firewalls must have 24/7 capability. Isolate process control systems from remote access. Delete the unused service provisions and remove unused ports connections in the operating systems to prevent unauthorized use in network based attacks. Network based attacks can further be avoided by hardening of process control systems.

Demilitarized Zone (DMZ): DMZ is a good technique for securing communication network based on the principal of network separation strategy between secured network (SCADA, DCS LAN) and Corporate Intranet, which can be further separated with another DMZ from internet if connected. Double protection is provided by the use of two DMZ, first between SCADA system and Corporate System and second between Corporate Intranet System and Internet. DMZ is a zone between an inner firewall and an outer firewall, ensure that server and database resides in safe place. The firewall acts as a filter that permits the data to enter from selected ports and blocks others. It is properly configured to protect passwords, IP addresses, and files. At DMZ output a router is used as border router to route information to correct destination.

A Simple Secured SCADA System

A simple secured SCADA system will comprise of a Corporate Server supported by Data Historian, Work Station, and a Printer in minimum configuration but protected by a Router and Firewall from a possible attack from Internet connection. Control Server supported by Human Machine Interface (HMI), Engineering Work Station in minimum configuration is protected by another set of Firewall and a Router in between Corporate Business systems, as shown in Figure 1. PLC's monitoring pollution sensors, temperature sensors, noise level sensors, water level sensors, flow rate sensors, pressure sensors, oil level sensors etc and controlling actuators of solenoid valves, pressure

regulators, servo drives, variable frequency drives, temperature regulators, humidity regulators etc are connected to Remote Terminal Units (RTU), which are in torn connected to Control Server through Satellite, Power Line Carrier Comm. (PLCC), WAN Network, Microwave/ Radio Comm., Cellular comm. via Front End Processor (FEP), as shown in Figure 2.

A LARGE PROCESS SECURED CONTROL SYSTEM

Large Process Control Server may be supported by Data Historian, Data Acquisition Server, Engineering Workstation, Database Server, HMI Computer, and Configuration Server, all connected in Control System LAN configuration. Corporate Business Server is supported by Web Application Server, Workstation, Authentication Server, FTP Server, DNS Server, and email Server; all of them are connected through Corporate Firewall to Internet Infrastructure.

Control System LAN is protected from cyber attacks, by using control system Firewall between corporate LAN and control LAN and creating DE-militarized Zone (DMZ) for Authentication Server, Security Server, and Data Base Historian, as shown in Figure 3. All field process devices i.e. Intelligent Electronics Devises (IED's), PLC's, RTU's and Controllers are connected to Control Server through Modem Pool and Communication Interface Infrastructure.

Firewalls must be configured with tightly rule bases, should have regular reviews, be managed by trained administrators, changes managed under strict control. Management and monitoring of firewalls must have 24/7 capability. Isolate process control systems from remote access where possible. Remove TCP/IP connections between safety systems and process control systems. Ensure all inbuilt system security features are enabled. Hardening of process control systems to prevent network-based attacks must be done. Remove unused services and ports in the operating systems and applications to prevent unauthorized use.

INTRUSION DETECTION SYSTEMS

The intelligently monitoring the events and analyzing the signs of violations of the security policy in a computer based control system or network is the process of intrusion detection. While designing IDS, the security issues of Availability, Utility, Authenticity, Confidentiality, Integrity, and Possession of a computer or network must be considered. (Alsiherov, F., & Kim, T. 2010), by NIST (Matsui, M. 1993), and (NIST 2013). . Intrusion Detection Systems (IDS) are software or hardware products that automate the monitoring and analysis process. IDSes are of either Signature Based IDS (SBIDS) or Anomaly Based IDS (ABIDS) as explained (Hamalainen, P., Alho, T., Hannikainen, M., & Hamalainen, T. D. 2006). Signature of known attacks is stored in SBIDS and the events are matched against the stored signature, it will indicate an intrusion if a match is found. The new attacks cannot be detected since its signatures are unknown. (Wolkerstorfer, J., Oswald, E., & Lamberger, M. 2002) and in ABIDS, it can detect unknown attacks as well as "zero days" attacks. Data mining based detection, machine learning based detection, knowledge based detection and statistical anomaly detection are the techniques used in ABIDS for attack detections. In ABIDS, the behavior of the system is monitored, if it deviates significantly from the expected behavior and parameters exceeds the threshold value then it is termed as attack. (Lamberger, M., et al 2009).

Passive Mode Security Implementation

In passive mode security, the device is connected to a hub or switch on the link between the SCADA master control station and the nodes it controls between Utility Intranet and SCADA networks. The intrusion detector sniffs packets as they pass by, saves a copy to analyze, and generate alerts if a security breach has taken place. As shown in the Figure 2, the security device is not in line with the communications, so a failed security device does not block the communications link. The security device can only report malicious packets it detects, but it cannot block from reaching intended recipient.

Network Intrusion Detection System

A researcher Vern Pexson of Lawrence Berkeley National Laboratory developed a system named Bro for detecting network intruders in real time. Bro is an open source UNIX based network-monitoring software, which can be used to develop a Network Intrusion Detection System, by collecting network measurements, traffic base lining and conducting forensic investigations. It has two layers, first layer for BRO Event Engine for analyzing live or recorded network traffic or trace files to generate neutral events, which reduces a kernel filtered traffic stream into a series of higher-level events.

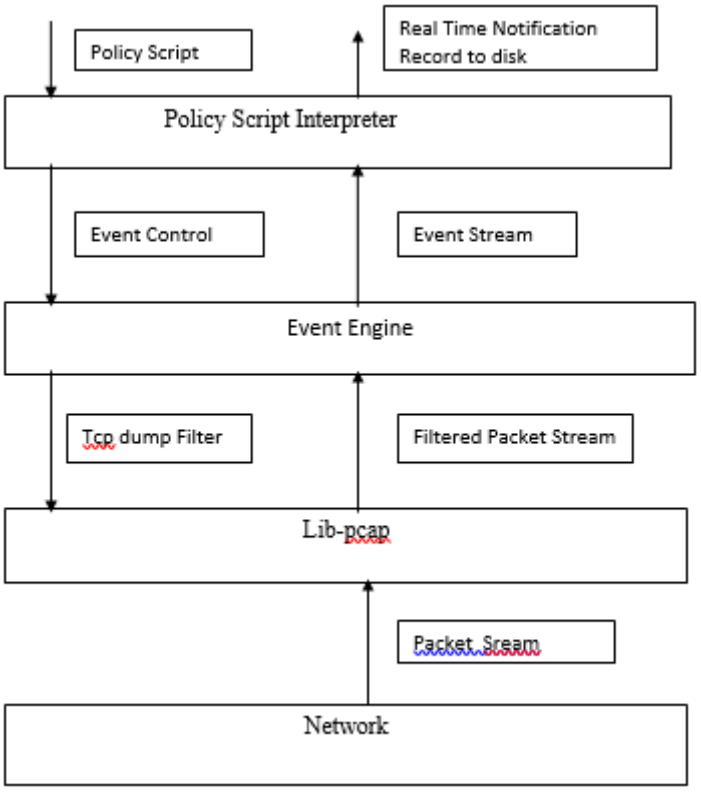

Figure 2.4 Structure of the BRO System

Data Mining Based Intrusion Systems

This architecture can carry out data gathering, analyzing data, archiving data, sharing, distribution and model generation. The sensor data may be continuous or discrete, symbolic or numerical. The system has independent of sensor data format, which may have arbitrary number of features. A model may be from a neural network, to a probabilistic model or to a set of rules. An XML encoding is used to enable heterogeneity and allow each component exchange data or

models. Common Intrusion detection Framework (CIDF), IDSs and IDMEF are used to ensure exchange attack information in standard formats to detect distributed intrusions collectively. The main advantage is the high performance and scalability achieved. All components may be in the same local network and workload is shared among its components. If the components are kept in different networks then collective collaboration with other IDSs in the Internet is used.

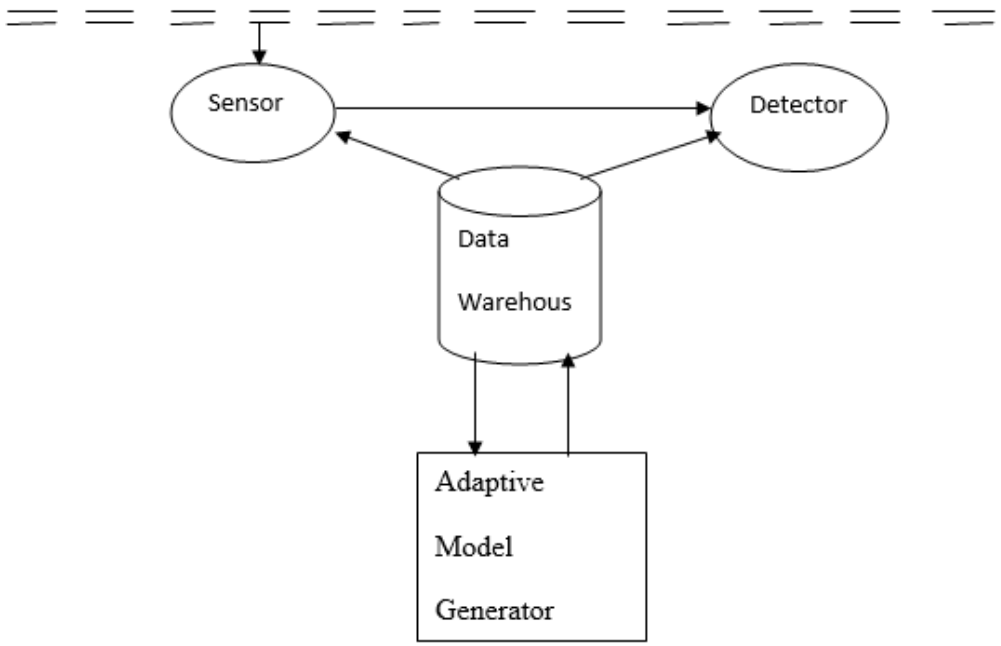

Figure 2.5 Data Mining based IDS

Sensors record raw data from monitored system detect and form features of model evaluation. A framework of a Basic Auditing Module (BAM) is formed by using multiple sensors data. The features are extracted from raw data in BAM and these are encoded in XML. Detectors use detection models to evaluate and process sensor data to decide if it was an attack or not and send the results to the warehouse for records and analysis. Multiple detectors may be deployed to analyze different events in parallel. The high-speed front-end detectors perform quick intrusion detection for high volume traffic. Back end detectors perform through analysis, which consume more time.

Data warehouse has a centralized storage for models and data. It uses relational database features with a provision of stored procedure calls for complicated calculations to carry out automatic data sampling on the server. The data and results from multiple sensors from different IDSs are collected over a longer period; it helps in the detection of complicated and large-scale attacks. Model Generators facilitate the rapid development and updated intrusion detection model distribution. When the attack is detected first time as an anomaly, its exemplary processed data by model generator will match intrusion data sets in the warehouse and it will automatically generate a model to detect the new intrusion, and to distribute the model to other detectors. Prototype of data mining and Common Intrusion detection Framework based Intrusion Detector system can be implemented.

Comparison of Snort and Bro Open Source Network IDS

The critical data of consumer specific information for financial services of the industry are to be protected from attackers. Enemies and competitors must protect the physical vulnerable data of floods, earthquakes, electric blackouts and hurricane disasters from strategic exploitation.

INTRUSION PREVENTION SYSTEMS

A Network Intrusion Detection System (NIDS) is an ID, which identifies malicious actions such as denial of service attacks; port scans or even attempts to crack into computers by monitoring network traffic. A host based IDS's are system monitors and analyses the internals of a computing system rather than the network packets. There are several open source NIDS, such as Bro, Snort, Shadow, Shoki, Spade, M-ice, Firestorm, etc. Bro IDS is confined to UNIX operating system, has the ability for multi-layer analysis, behavioral monitoring, policy-based intrusion detection and logging network activity. Bro IDS has the ability to run in high-speed networks, is very effective and able to capture from GBPS networks, and is suitable for large-scale networks like SCADA systems for Petroleum Industry. (Hamalainen, P., Alho, T., Hannikainen, M., & Hamalainen, T. D. 2006) and for Electric Power Grids (Kaur, A., Bhardwaj, P.,

& Kumar, N. 2013). Bro based IDS can be designed to meet SCADA specific security requirements for DNP3 Protocol (Lamberger, M., et al 2009).

A real- time network traffic analyzer can be designed to generate events for proper operation of the SCADA system as per designated to build in parsers, to detect and indicate violations from defined security policies, as per intrusion detection policy decisions of network traffic. The abnormal communication patterns observed from validation policies due to replayed network packets, mal formed packets may indicate denial of service attacks, device failures or system miss- configuration, malicious operation etc. After systematic processing and analyzing suspicious network traffic, we must correlate semantic related control decisions for proper selection action in the control Center. On proper attack detection and accurately identifying attack, it becomes easier to prevent intrusion in the system. Network firewalls control data flow between control server and corporate systems. NIST SP 800-41, Guidelines on Firewalls and Firewall Policy, which provide general guidance for selection of firewalls and firewall policies, while designing SCADA security. The firewall should have features of extensive logging of events, De-Militarized Zone (DMZ) based policy routing, IDS, access list, etc. Deployment of firewall use is to have effective intrusion prevention policy based on network topology.

The intelligently monitoring the events and analyzing the signs of violations of the security policy in a computer based control system or network is the process of Intrusion detection. While designing IDS, the security issues of Availability, Utility, Authenticity, Confidentiality, Integrity, and Possession of a computer or network must be considered. Intrusion Detection Systems (IDS) are software or hardware products that automate the monitoring and analysis process. IDSes are of either Signature Based IDS (SBIDS) or Anomaly Based IDS (ABIDS). In SBIDS, signature of known attacks are stored and the events are matched against the stored signature, it will signal an intrusion if a match is found, but it cannot detect new attacks whose signatures are unknown. In ABIDS, it can detect unknown attacks as well as “zero days” attacks. Techniques like statistical anomaly detection, data mining based detection, knowledge based detection and machine learning based detection is used in ABIDS. In ABIDS,

the behavior of the system is monitored, if it deviates significantly from the expected behavior and parameters exceeds the threshold value then it is termed as attack.

Distributed Network Protocol (DNP3)

It is a set of communications protocols used for controlling components in process automation systems, or in utilities such as Electric Power Grid Applications and Water companies. It facilitate in communications between various data acquisition and control systems. SCADA system is used in Control Centers, Master Stations, Intelligent Electronics Devices and Remote Terminal Units. Network Topologies of DNP3 may be Multi-drop, Hierarchical, Direct one to one or multiple Master types.

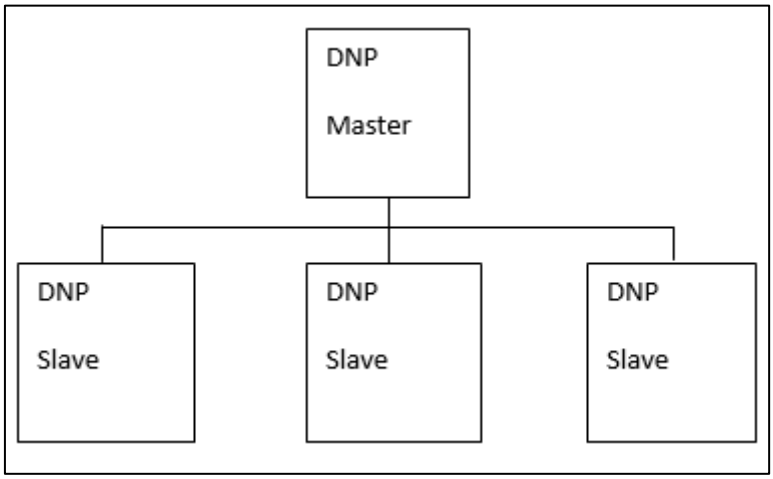

Figure 2.6 Multi-drop Configuration

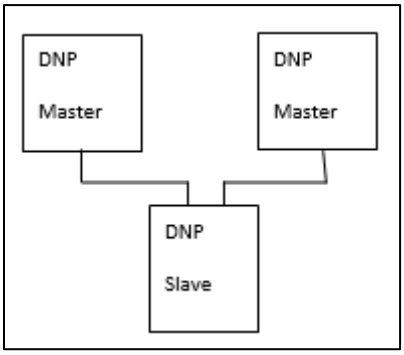

Figure 2.7 Multiple Master

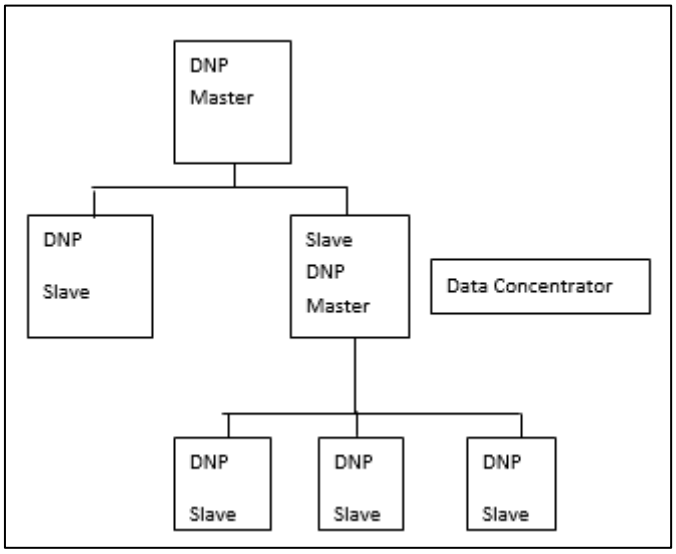

Figure 2.8 Hierarchical Configuration

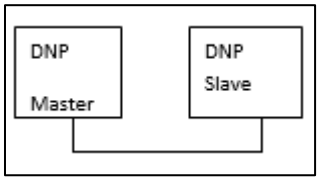

Figure 2.9 One to One

INDUSTRIAL CYBER SECURITY AND HUMAN MACHINE INTERFACE IMPLICATIONS

The productivity can be increased, downtime reduced of the industry by efficiently using SCADA with appropriate Human Machine interface and high-end network systems. The parser of the DNP3 generates the events of the SCADA system. The semantics of the events are sent to the proper event handlers. As per Bro scripts the event handler are defined and semantic of each event delivered to the corresponding event handler. The interpreter of the policy

script executed the script for producing results and analysis to generate signal of abnormal activity in the network and sound alert.

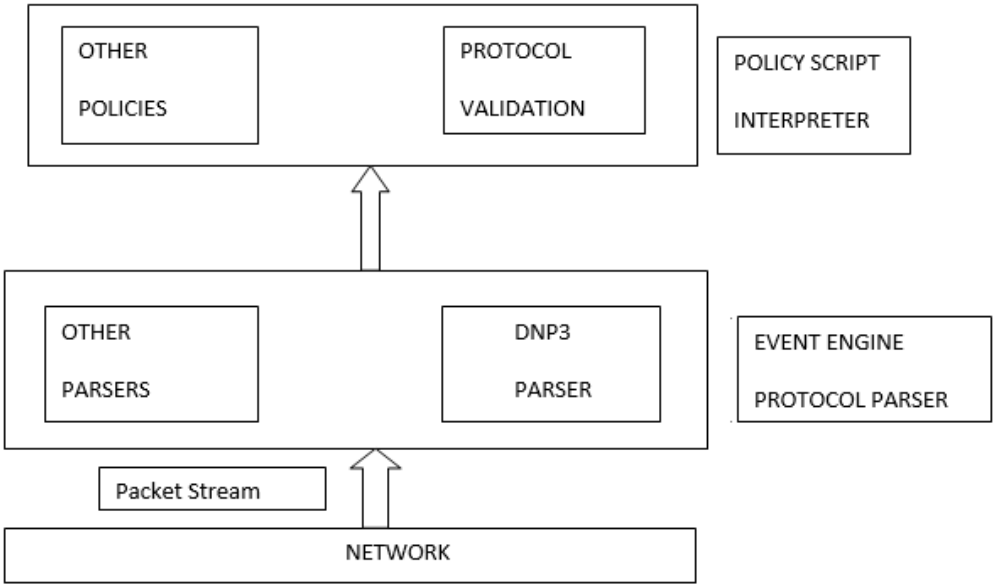

Figure 2.10 DNP3 Analyzer Based on BRO

NP3 Parser: The network packet parser decodes byte streams into meaningful data field as per protocol definition. In DNP3, parser has compiler-assisted TOL named as binpac to reduce the development time and to ensure proper correctness. The binpac scripts are designed to represent hierarchical structure of the network protocol. The binpac scripts are automatically is translated into C++ and integrated into Bro.

Event Loggers: The various critical events takes place in industrial processes of SCADA and complete audit trials are conducted for its proper analyses and data of time and date stamping of these events is recorded in the Event Loggers. In some critical processes, data is generated in milliseconds and corrective action has to be taken immediately to keep losses to a minimum level. The monitoring and control of interfacing devices special care has to be taken for generating time stamping by using time synchronizing source of satellite receiver of the Global Positioning System controlled SCADA system.

Event Handlers: The event handler to analyze the various network events parses the received DNP3 network packets. The semantic of the event is extracted to generate DNP3 request for operation of control Relay output block

eventdnp3_crob. The event handler is takes the necessary action corresponding to the data extracted from packet for the type of the operation needed and its duration. The interface between parser and policy script interpreter is controlled by the name and arguments of the event handler. The value of the arguments is generated during run time of parsing which update the semantic information of the events.

Protocol Validation Policy: Since the protocol use 37 combinations only out of 256 values for function code representation, so 8-bit integer is used for coding. The link layer header has ‘length field’ to represent for the following payload. The field length value should be verified with real payload length to find out its value and analyzed to detect attacks if found out of order of the range. The inner packet validation is checked to find dependencies between various data fields in a network packet. The protocol validation is for checking dependencies of intra and inter packet of various fields.

Verification of Protocol Validation Policy: It is defined on the context of SCADA systems operating Process Control Systems or Electrical Power Grids. Event handlers should be defined for Implementation:

dnp3 _app-request_ header

dnp3_app_response_header

dnp3_object_header

These event handlers extract values of the function code, the object type and other semantic information from the DNP3 request / response headers and object headers. The object with the group number 12 and variation number 1 describe a CROB (Control Relay Output Block) object. This object can be initiated by requests with function codes 3, 4, 5, and 6. Bro scripts for validate the rule will be.

If ((Obj_Type = 0x0c01) && ((Fun Code <0x03) || (Fun Code > 0x06)))

ALERT …. ;

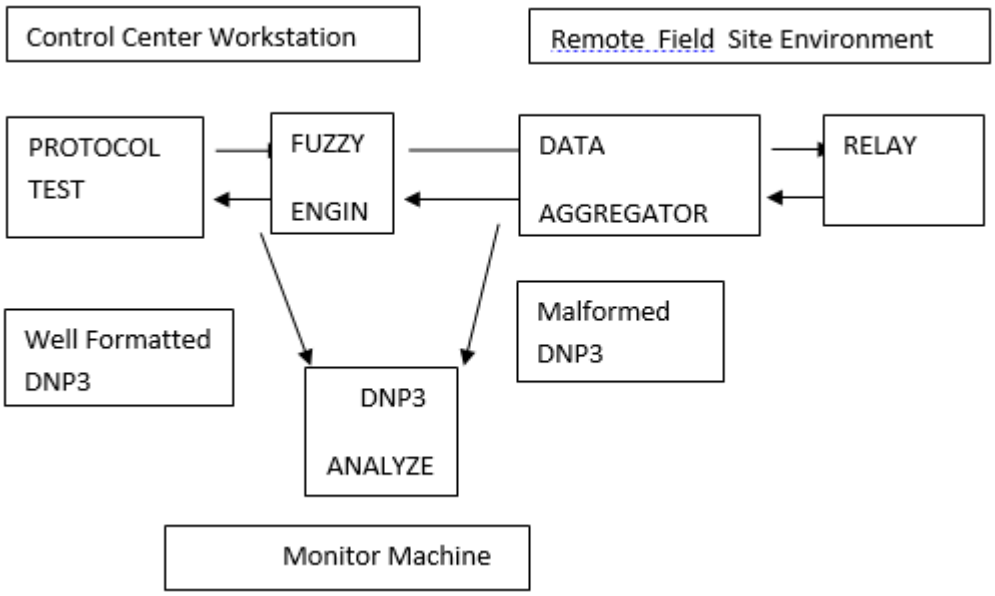

Figure 2.11 Simulated SCADA Test-Beds

Performance Evaluation: A packet of trace of 1GB is generated in the Analyzer for analyzing throughput of DNP3 for the packets processed in each second. The packet trace has TCP packets for opening and closing communication sessions, well-formatted packets, and may have malformed packets. The process control industry utilize DNP3 analyzer to first passively to find out its attack monitoring results and then to process network packets in real time to control process performance. The test results after performance confirms its analysis and monitoring capability in a real SCADA system.

Modern SCADA Systems: Modern SCADA Systems integrate management plant floor with enterprise resource planning (ERP) software, and product lifecycle management be carried out (PLM) of design solution software and manufacturing execution systems (MES).Optimization of resource planning will enhance efficiency by seamless exchange of ERP and PLM intelligence. Improvements in manufacturing must in order to obtain maximum efficiency. The process planning and product innovation are carried out by integration of MES and PLM. It will enable continuous improvement in production cycle. The SCADA System allows equipment located within factory premises, at different locations in a city, at different states of the c country. The system must provide full process automation and applications. The trends will shape the design and

application of all automation and control solutions in scope and applications of SCADA systems. Bi-directional exchange of recommendations between MES and PLM helps in continuous improvement throughout production cycle.

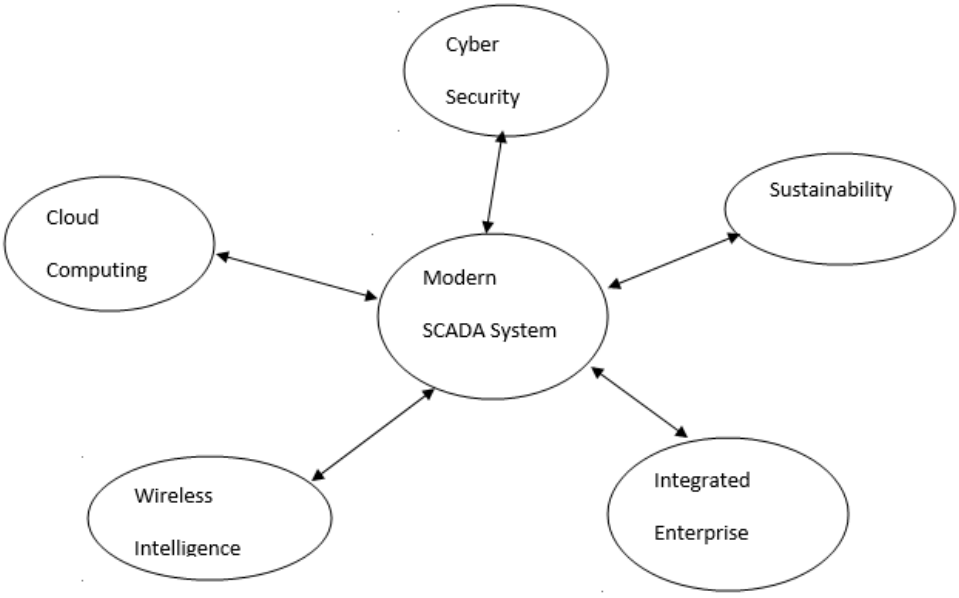

Figure 2.12 Modern Industrial Automation SCADA

Integration of MES, PLM, and ERP software will develop enterprise ecosystem and synchronized business strategy to enhance effectively efficiency. Wireless intelligence of Wi-Fi, RFID and Wi-Max is going to gain greater use in IT infrastructure and services. Real –time video surveillance, wireless data networking for system intelligence to make well-informed process decisions effectively and efficiently.

The cloud computing is likely to change the face of data storage and influence business decisions. SMART clouds with enhanced security will be a technical standard. A hybrid cloud strategy of public and private clouds for benefit of security from cyber threats. Robotics principles for performing some complex manufacturing processes are likely to be used to deploy robots for reducing material consumption and improving product quality. A wireless networks supporting a highly automated production process, interlinked seamlessly with enterprise software working through the clouds.

Future SCADA Systems must provide the provision of Industrial Cyber Security: SCADA and Human Machine interface enabled to reduce downtime and increased productivity, by using high end network capabilities Cyber threats and attacks are basically aimed at disrupting Industrial activity, political factors and competitive to drive some benefits spread across monetary . These attacks disrupt production process and impact quality of produced goods, which may affect the reputation of industry.

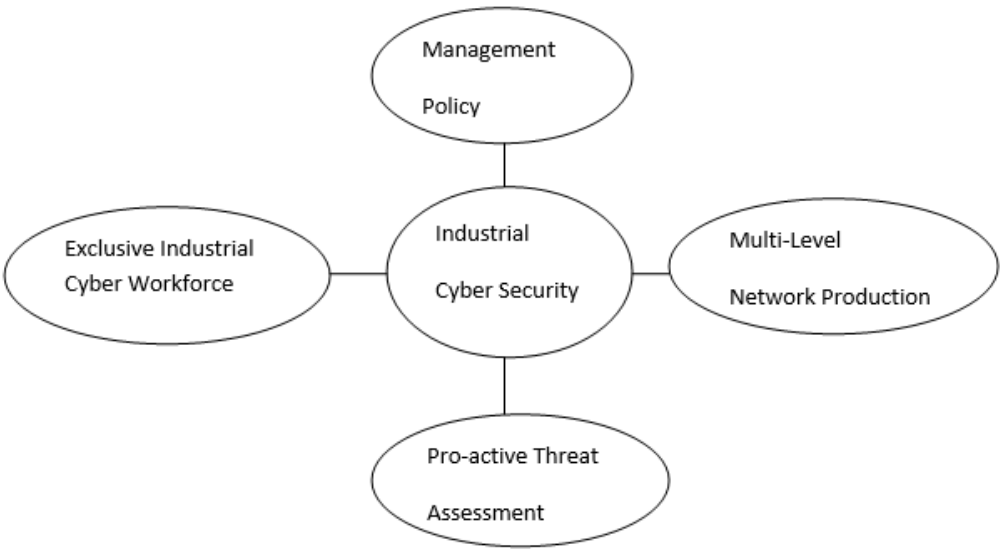

Figure 2.13 Cyber Threats and Strategy

A number of high –profile SCADA attacks in recent past indicate that cyber security mechanisms play a decisive role in the selection of SCADA products from various automation vendors. A Cyber security policy formulation needs to be planned. The establishment of an industrial cyber work force, pro-active threat assessment implementation planning is needed, and multi-level network protection implementation to be established in the Organization.

Integrated Enterprise Ecosystem: The generation of comprehensive business intelligence, by means of integration of PLM, MES with ERP software, is accessible within the enterprise. MES domain plays a pivotal role in generating enterprise ecosystem. The high-end communication capabilities built within the SCADA architecture involving higher layers of enterprise software. It will possible to bridge the gap between the business process application and plant

floor. The design pursuit need long term strategy, which helps automation vendors sustain their leadership in competitive markets.

GLOBAL SCADA SYSTEMS

There is a high growth across different domains of SCADA end user sectors. It is expected to grow at a compound annual growth rate (CAGR) of 8 % in the near future. Water and Wastewater, power, Gas and Oil Industries are the key Industrial sectors using SCADA solutions. Automation vendors are interested in market sustenance and profitability and will emerge with new innovative solutions for growing needs of end user, in terms of increased efficiency and improved profitability.

Cyber Security Cloud Computing Sustainability Modern SCADA System Wireless Intelligence Integrated Enterprise Commented [A2]: Figure 2.12 Modern Industrial Automation SCADA Integration of MES, PLM, and ERP software will develop enterprise ecosystem and synchronized business strategy to enhance effectively efficiency. Wireless intelligence of Wi-Fi, RFID and Wi-Max is going to gain greater use in IT infrastructure and services. Real –time video surveillance, wireless data networking for system.

Robotic principles for performing some complex manufacturing processes are likely to be used to deploy robots for reducing material consumption and improving product quality. A Cyber security policy formulation needs to be planned, while using SCADA products from various automation vendors. The establishment of an industrial cyber work force, pro-active threat assessment implementation planning is needed, and multi-level network protection implementation to be established in the Organization. Integrated Enterprise Ecosystem: The generation of comprehensive business intelligence, by means of integration of PLM, MES with ERP software, is accessible within the enterprise. MES domain plays a pivotal role in generating enterprise. Future SCADA Systems must provide the provision of Industrial Cyber Security and Appropriate the Enterprise Ecosystem. It will be possible to bridge the gap between the business process application and plant floor, by involving higher layers of enterprise software. The design pursuit need long term strategy that

helps automation. The firewall use is to have effective intrusion prevention policy based on network topology. Specification based Intrusion Detection System.

CHAPTER – 3

AES

CHAPTER – 3 AES

AES IMPLEMENTATION SCHEMES

The AES algorithm may be implemented by following schemes for secured data communication.

a) Software Schemes: Software program schemes are easier to implement, low in cost. The processing speed is slow and has a low physical security, has high chances of being corrupted by viruses. The high volume data transmission at high speed in secured communication with proper physical security demands hardware implementation of cryptography.

b) VLSI /ASICS Schemes: This ASICS design schemes have very high development costs and require long development time, however the cost per chip may be low if quantity produced is very high, which is normally low for security devices. The flexibility in design variation is not available. Researchers (Mangard, S., Pramstaller, N., & Oswald, E. 2005), have tried an implementation of S-Boxes of AES in the past to make compact cryptosystem. The optimum design for AES S-Boxes has been attempted by researchers (Uskov, A., Byerly, A., & Heinemann, C. 2016) proposed construction of Optimum CFA for Compact High-Throughput AES S-Boxes, using CFA with isomorphic mapping, which results in the minimal implementation area. The optimization of CFA combinatorial circuit in the field of mapping, basis representation, field polynomials and isomorphic mapping helps to identify a short critical path for VLSI architecture implementation.

c) FPGA Schemes: FPGA implementation schemes have low development cost and requires less development time. The flexibility in design variations is available if required in implementation stage; the security may be moderate to high. The developmental time is low and marketing time is short.

The research proposal will deal with an FPGA implementation of AES encryption/decryption with key size of 256 bits, simulation, synthesis reports will be generated, and the results will be compared with the implementations done in the past by other researchers. Our research proposal will have key expansion module to generate round keys calculated as per the general guidelines. Our proposal is to use lookup table approach implementation for S-box to obtain high throughput by data pipeline for all rounds to achieve low latency as well.

FPGA IMPLEMENTATION OF AES WITH 128-BIT SECURITY KEY

Plain text data of 128 bits is encrypted using 128 bits round key in 10 rounds as shown in Figure 3.1 on left side and cipher text data is decrypted using the same set of round key but using in reverse order for decryption. For data encryption operation, in round one to round nine we perform BS, SR, MC, and AK transformation during each round and in round tenth MC transformations not included. For data decryption operation, the reverse order of rounds is followed. We perform inverse SR, inverse BS immediately after initial AK transformation using round key 10. During remaining nine decryption rounds, the same order of inverse transformations is used, but including inverse MC transformation in the beginning of the every round with round key number in reducing order. After last of AK transformation, we get original plain text output data.

The input secret key of 128 bits is expanded into key for ten rounds of 128 bits each. The 128 bits secret key expansion operation is shown in Figure 3.2. Round key0 is used for first AK operation with plain text data during start of encryption. Round key1 is used for AK operation during round1 of encryption. Round key2 to round key10 are generated for AK operations, for rounds 2 to 10 as shown in the Figure 3.2. Round keys generated during encryption are stored and utilized for AK operations of decryption also but are used in reverse direction.

When start pulse is given to the controller module, clock pulse, reset pulse, enable pulse and en/de pulse are generated by controller module. Controller module sends first reset and clock pulses to key generation module and encryption / decryption module, then send 0/1 signal to encryption/ decryption module for encryption or decryption operation depending signal level is 0 or 1 respectively. The input security key of 128 bits and input plain text / cipher text of 128 bits data are entered in key generation module and encryption / decryption module, respectively, on getting enable pulse from controller module as shown in Figure 3.3. The encrypted/decrypted data of 128 bits is outputted at output port, and done pulse is generated by encryption/decryption module. Simulation results in Figure 3.4. Comparison of results with reported work in Table No. 3.1. (Next Page).

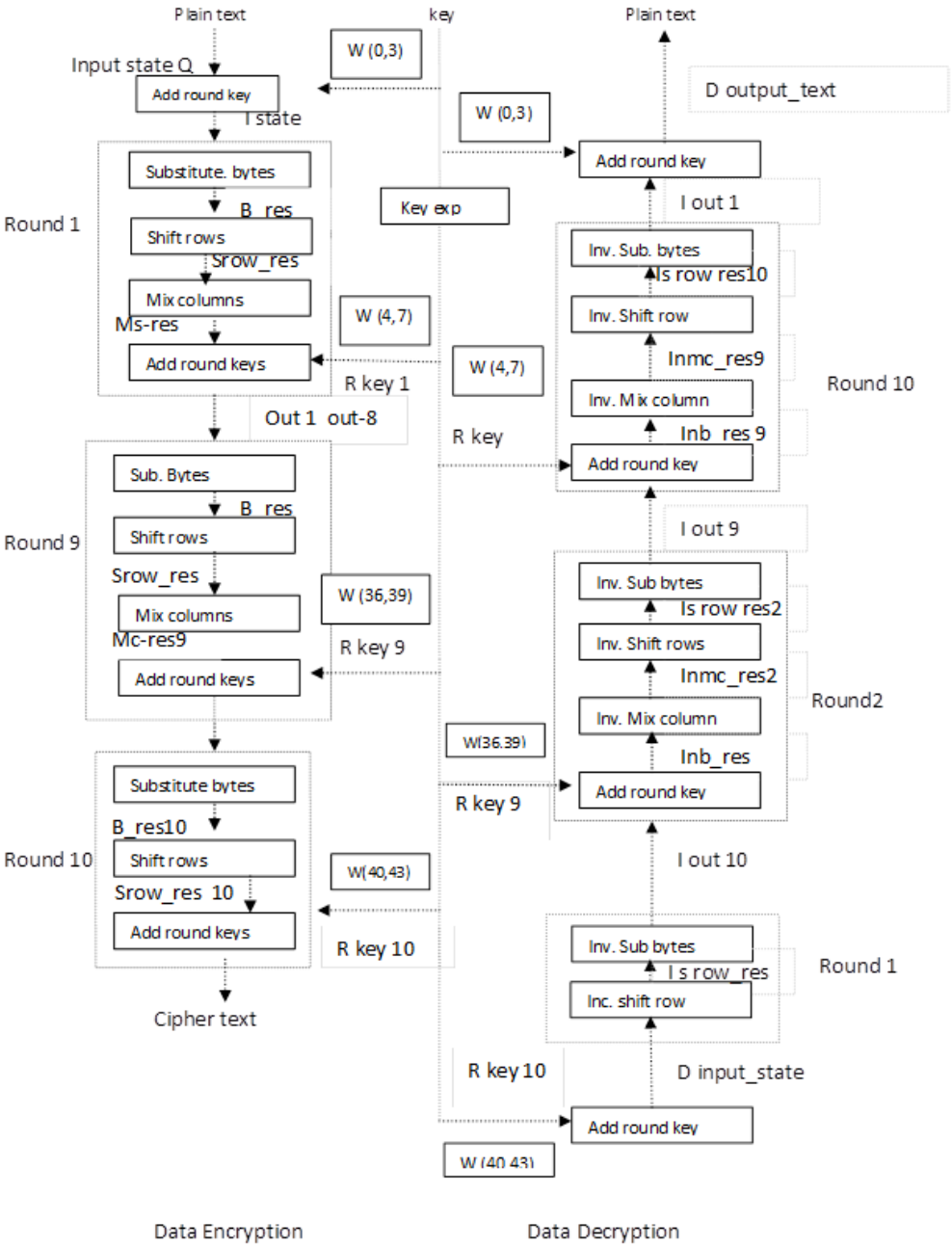

Figure 3.1 Data Encryption and Decryption -128 Bits

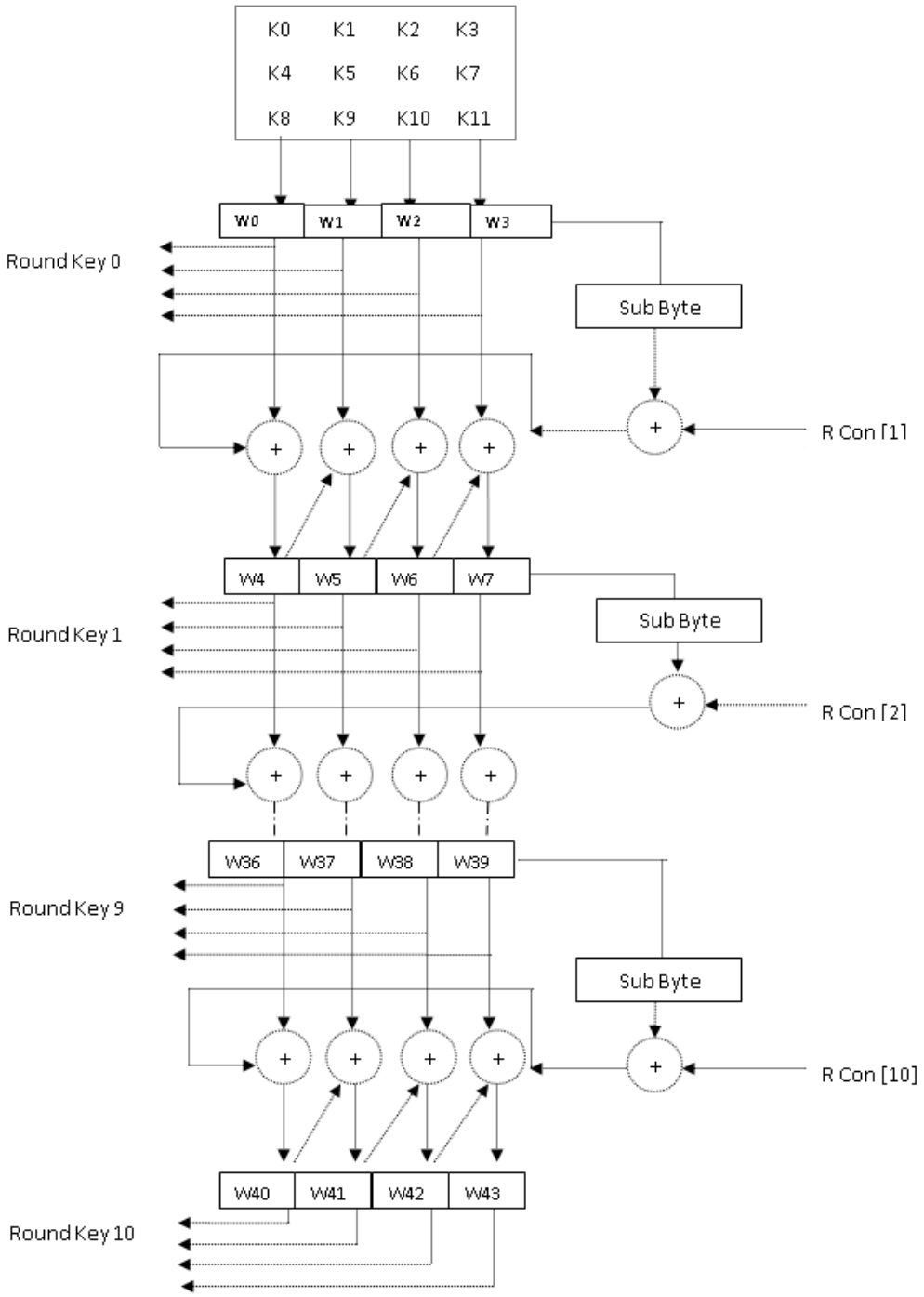

Figure 3.2 128 bits Security key expansion operation

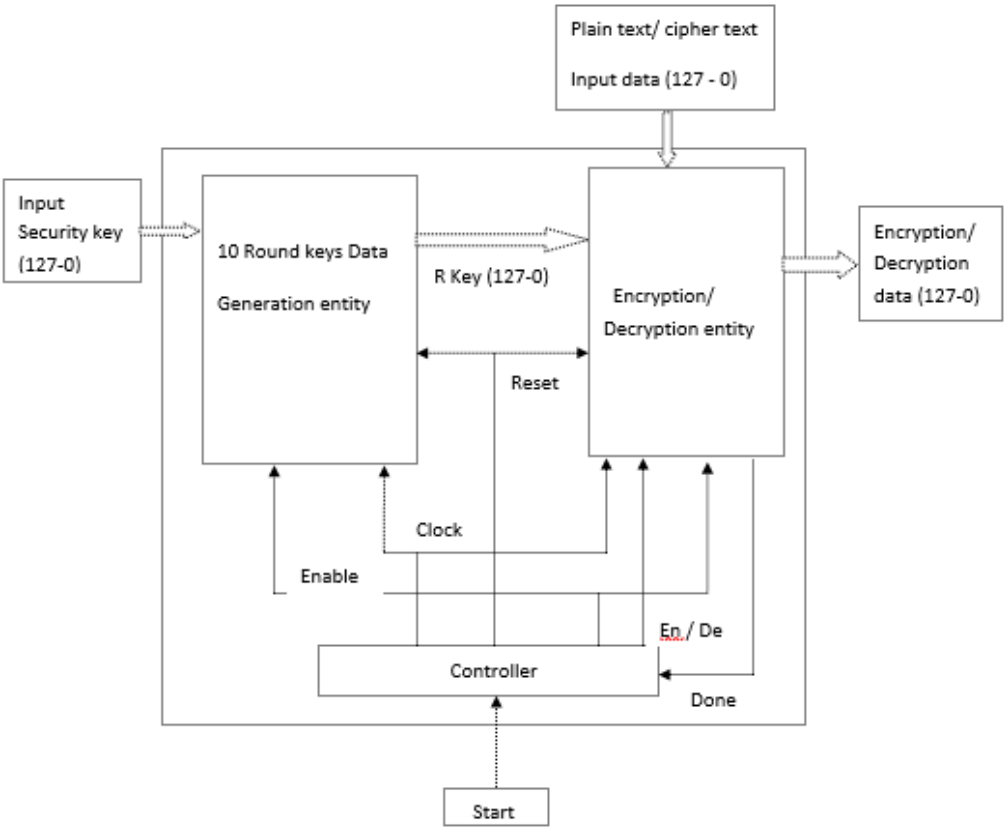

Figure 3.3 Encryption and Decryption top Level Entity for 128 bit key.

Table 3.1 Comparison of results for AES with 128 bits security key

Design	Device used	Area/Slices used	Throughput Megabits/sec	Throughput Megabits/Slice	Maximum frequency in MHz
1. K. Gaj & P. Chodowiec [28]	XCV1000BG560-6 XC2S30-6	2902 222; GRAM-3	331.5 166	----- 0.132	----- 60
2. Dandalis [ ]	XCV-1000	5673	353.0	0.062	-----
3. Elbirt et.al [41]	XCV1000-4	10992; BRAM-0	-----	-----	31.8
4. Mcloone	XCV812E-8	2000; BRAM-224	-----	-----	93.3
5. Helion	Virtex 4-11	1016	-----	-----	200.0
6. G. Rouvroy [84]	XC3S50-4	163 BRAM-3	208	1.26	71
7. Swinder Kaur [56]	Virtex2 p-7	6279; BRAM-5			119.95
8. Amandeep Kaur[55]	XC2VP30-5-FF896	1127	-----	-----	247.3
9. Thulasimani [85]	XC-2V600BF-957-6	2943	666.7	0.226	-----
10. Our Design AES-128 bits security Key	XC5VTX240T-2FF 1759-2	10240; BRAM-0	**4720**	**0.460**	472.8

SIMULATION AND SYNTHESIS RESULTS OF 128 BIT KEY

The design has been coded using VHDL, all the results are synthesized based on Xilinx ISE Software 12.4 version, and target device used was xc5vtx240t-2-ff1759. The results of simulation of encryption/decryption with security key of 128 bits with 128 bits input data, all 128 bits of “one’s” value are shown in Figure 3.4. We find encrypted data at transmitter output as quite in random order, since AES algorithm ensures good dispersion and confusion of transmitted data. Simulation results also show that input plaintext data is properly ciphered in encryption operation and when ciphered text is given as input to decryption operation, deciphered data is found to be the original input data of encryption operation. All the round keys generated during encryption operation are found to be the same as given in NIST documents for security key of 128 bits. Simulation results with all input data as “zeroes” are shown in Figure 3.5.

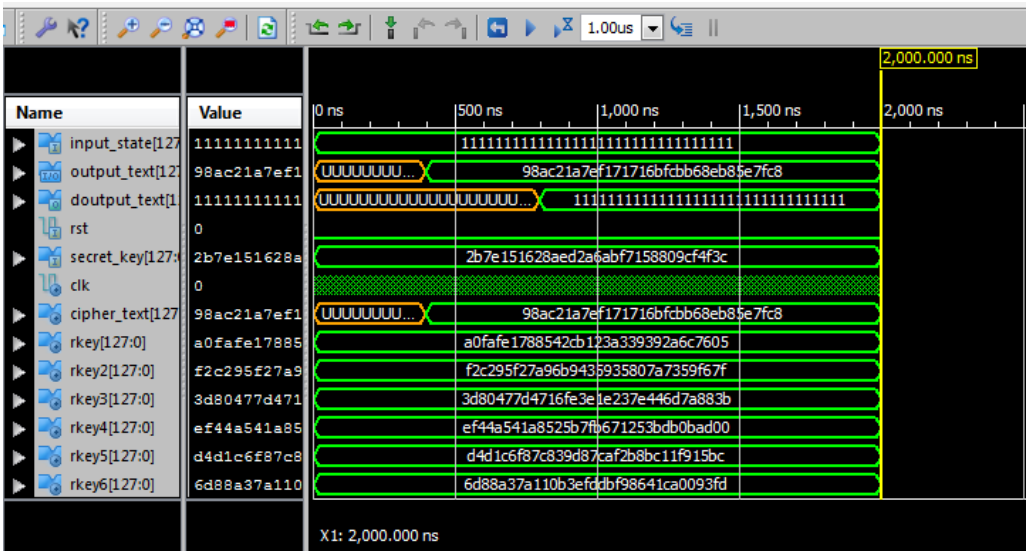

Figure 3.4 Simulation results with all the 128 input data bits as “ones”.

Synthesis reports for 128 bits security key are generated for AES algorithm based on Xilinx ISE software 12.4 versions for target device xc5vtx240-2-ff1759 are generated. Synthesis report data generated is given below.

1. No. of ROMs: 360
2. No. of Flip Flops: 10240
3. No. of input and output pins: 515
4. No. of Slice LUT’s: 19974
5. Clock period: 2.115nS
6. Maximum Frequency: 472.82 MHz
7. Delay: 2.115nS
8. Throughput: 64 GBPS

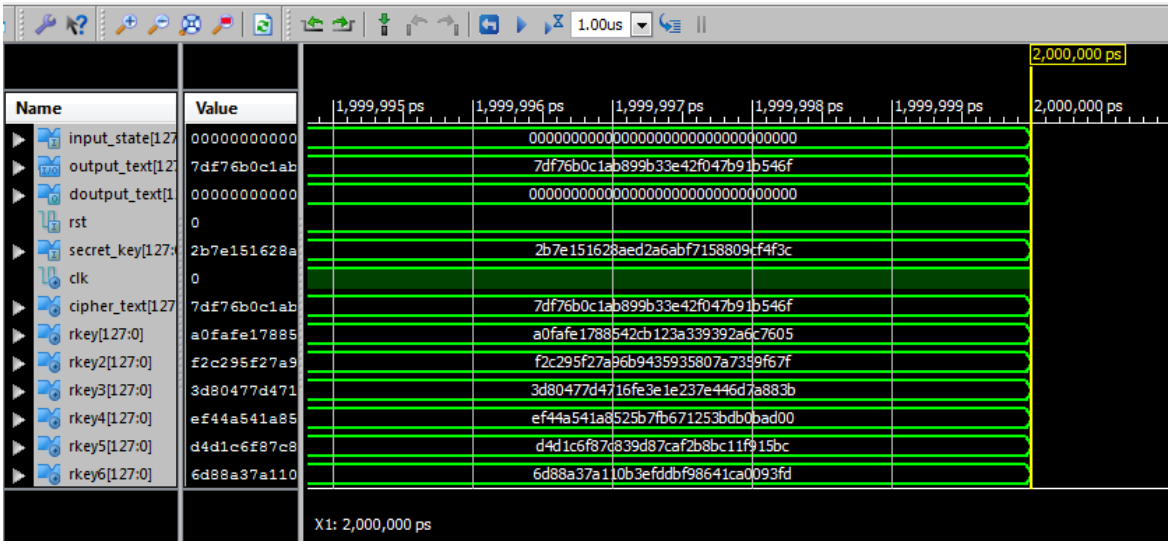

Figure 3.5 Simulation results with all the 128 input data bits as “zeros”

DATA PERFORMANCE OF AES CIPHERS

The performance analysis of AES Cipher in single core architecture have been studied, in various modes of operation like, Electronic code book (ECB), Counter (CTR), Cipher feedback (CFB), Cipher Block chaining (CBC), and Output feedback (OFB). The performance of secured data communication of rich multimedia data through a VPN over IPSec Protocol depends on the efficiency of encryption algorithm, data integrity, authentication algorithm and mode of cipher operation. The concept of using higher frequency of the CPU from low frequency for the frequency scaling, the concept of using multi-core computer architecture in place of single processor for processing data has taken place, so the number of processing cores are added in various new process architectures. Some of the performance improvement scope is provides using new set of processor Instruction sets and providing scope by means of software programming also.

Some manufacturers to increase the computing power of processor architecture and using new set of processor instruction sets also try the hardware accelerator implementation. A set of new Instructions set by Intel for MP AES-NI CPU processor has improved the processing speed as compared to CL R7 250 d GPU

and CL-HD7540 iGPU processors. These instructions are designed to carry out complex and computationally better steps of AES algorithm, which in turn accelerate the execution of AES algorithm. AES-NI instructions improved the performance by a factor of 4 to 10 in comparison to complete software programmes.

New Instructions Set of AES-NI:

1. AESENC: It carries out encryption by processing all the 4 transformations in a single instruction of AESENC, by performing ShiftRows, SubByte, MixCoulumns and AddRoundKey, in one stroke.
2. AESENCLST: It combines ShiftRows, SubByte, and AddRoundKey transformations in another Instruction.
3. AESDEC: It performs InvShiftRows, InvSubByte, InvMixCoulumns and AddRoundKey, 4 transformations combined in another Instruction for decryption function.
4. AESDECLAST: It combines InvShiftRows, InvSubByte, and AddRoundKey, 3 transformations in another Instruction.
5. AESKEYGENASSIT: It is used to generate the round keys for encryption.
6. AESIMC: It is used for generating round keys for processing decryption from round keys used in encryption

AESENC	xmm1, xmm2/m128	AESENCLAST	xmm1, xmm2/m128
Tmp :=	xmm1	Tmp :=	xmm1
Roundkey :=	xmm2/m128	Round Key	xmm2/m128
Tmp :=	ShiftRows (Tmp)	Tmp :=	ShiftRows (Tmp)
Tmp :=	Subbyte (Tmp)	Tmp :=	Subbyte (Tmp)
Tmp :=	MixColumns (Tmp)		
xmm1 :=	Tmp xor Round Key	xmm1 :=	Tmp xor RoundKey

Figure 3.6 AESENC and AESENCLAST Instructions of AES –NI

AESDEC	xmm1, xmm2/m128	AESDECLAST	xmm1, xmm2/m128
Tmp :=	xmm1	State :=	xmm1
RoundKey :=	xmm2/m128	roundKey :=	xmm2/m128
Tmp :=	InvShiftRows (Tmp)	Tmp :=	InvShiftRows (State)
Tmp :=	InvSubBytes (Tmp)	Tmp :=	InvSubBytes (Tmp)
Tmp :=	InvColumns (Tmp)		
xmm1 :=	Tmp xor Round Key	xmm1 :	Tmp xor Round Key

Figure 3.7 AEDEC and AESDECLAST Instructions of AERS –NI

These Instructions of AESENC, AESENCLAST, AESDEC, and AESDECLAST are represented as pseudo code xmm1 and xmm2 of registers xmm. The grouped sequence of transformations of AES encryption and decryption is the longest sequence possible without the branch instructions, (Gyanchandani, M., Rana, J. L., & Yadav, R. N. 2012). These instructions have improved the performance in comparison of pure software implementations, having full flexibility of usability with all standard key lengths, standard mode of operations. These instructions have provided security enhancement also, by eliminating major timing and cache based attacks.

Overall x3.054 of AES performance with AES-NI hardware acceleration for all designated Ciphers. Increase of performance in CTR mode of x 4,294, in ECB mode by factor of x5.801. Researcher (Calomel.org 2015) analyzed the performance of AES Cipher with AES –NI Instructions in CBC mode and the results are given in Table - 2, testing specifications are as given below.

1. AES –NI Acceleration enabled
2. Libre SSL 2.3.0 (Open SSL 1.0.2d)
3. Free BSD 10.2.Clang LLVM compiler
4. 8192 byte blocks
5. Five test runs, the average speed reported

IMPACT OF AES-NI HARDWARE ACCELERATION ON IT SECURITY AND DATA PROCESSING PERFORMANCE

It has to be analyzed the performance verification with respect to

a) Various modes of AES Cipher operations

b) Streaming Rich Multi Media (RMM) Files and data

c) Data encoding/ decoding and data processing in IPSec VPN Networks

The data processing increase in AES performance in AES_NI enabled VS AES-NI disabled for each mode of AES operation analyzed (Thulasimani, L., & Madheswaran, M. 2010) reported and is given in Table -3.2.

Table – 3.2 Analysis of AES Cipher Performance with AES-NI enabled Mode and AES-NI disabled Mode on Intel Core i7-4790K Processor

AES-NI (Disabled or enabled	AES Cipher's Calculated Parameters PERF and Cycles	Modes CTR	Of ECB	AES CBC	Cipher CFB	Opera- OFB	tions Median Values
AES-Disabled	Mean PERF	335.600	280.690	274.453	273.110	261.20-5	284.972
AES Disabled	Median Cycles	11.919	14.261	14.574	14.646	15.314	14.143
AES-Enabled	Mean PERF	1441.142	1627.350	431.359	433.171	418.085	870.221
AES Enabled	Median Cycles	2.775	2.458	9.273	9.234	9.567	6.661
Increase	Of PERF	4.295	5.802	1.572	1.586	1.601	3.054

AES-NI hardware acceleration data performance of RMM in IPSec based VPN environment for AES-128 Cipher, in a real world on modern processors Intel Core i7-4790K in various modes

SECURITY ANALYSIS OF AES ALGORITHM

The data performance of in AES-NI architecture in ECB mode is highest, the drawback of this mode is that it encrypts plaintext blocks into identical cipher modes and it does not hide data patterns completely, so it has very low level of security.

The AES-128 cipher in CTR mode provides a satisfactory level of security and the second highest level of performance on AES-NI architecture, hence it is recommended for a secure transfer of data over VPN networks.

Improved performance through Web Server Consolidation

Intel-NI 6 instruction set accelerates the AES algorithm performance and guards it against memory pattern attacks. Web workload is a useful benchmark for checking the performance of Web servers used for secured data over Secure Socket Layers (SSL). Web workload uses encrypted long files using A Web server, client systems to generate load, a back end application server, PHP or java server pages (JSP) to generate dynamic web content.

Intel Xeon processor 5600 series-based servers and VMware VSphere by generating virtualization platform at a very low cost with improved performance and energy efficiency in IT service delivery of companies.

Set up for Web Workload Performance test using AES-NI

Intel Corp. analyzed the performance improvement by conducing test set for Web Workload using AES-NI instructions, consisting of 2 socket system configured with Intel Xeon processor X5680, 48GB memory and storage capacity of 15TB, details given in Figure . The performance improvement achieved details are given in Table -1. It was stated that AES-NI reduced computational overhead of encryption by 50 %, thus enabling 14% more users.

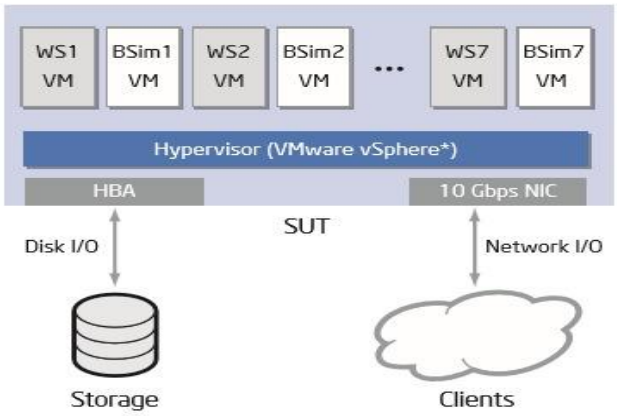

Figure 3.8 Test configurations for Web Workload

Figure 3.9 Web Workload Performances without SSL and with SSL and With SSL + AES-NI

It is stated that AES implemented using AES-NI performs better in efficiency and provides security against memory pattern attacks while software implementing of AES. The performance of AES-128 in CTR mode conducted using Intel Core i7-4790K processor with AES-NI acceleration disabled is equal to 335.600 MB/Sec and with AES-NI acceleration enabled obtained data 1,441.142 MB/Sec. The increase of AES performance due to AES-NI enabling

acceleration in CTR mode is equal to 4.294 times. A median value of increased AES performance from AES-NI disabled to AES enabled for all modes of operation and on various RMM test data sets is equal to 3.054.

HIGHLY FAST AND SECURE AES IMPLEMENTATION

The processing of AES transformations by conventional processors gets speed limited. The high speed intellectual processor cores (IP) dedicated processors with new long instruction sets have been developed by Intel Corporation, to accelerate the performance of Galois Field fixed field constant multiplication, an important element of AES algorithm, in comparison to pure software implementation speed. An instruction of Intel PCLMULQDQ for Intel Core processor can perform carry-less multiplication of two 64 bit operands, without propagation of carry values, by computing Galois Hash for efficient implementation of AES. A 127- bit output of two 64-bit operands is produced, which in turn may be used by software for generating the 255-bit output for GCM. The most significant bit equals 0 among 256 bit result.

Galois Counter Mode generates the message digest termed as Galois Hash from encrypted data meant for message authentication. The previous Galois Hash value is XOR- ed with the current cipher text block. The output is multiplied in GF (2128) with hash value. Irreducible polynomial g = g(x) = x128 +x7 +x2 +x +1, is used to produce GCM, as shown in Figure below.

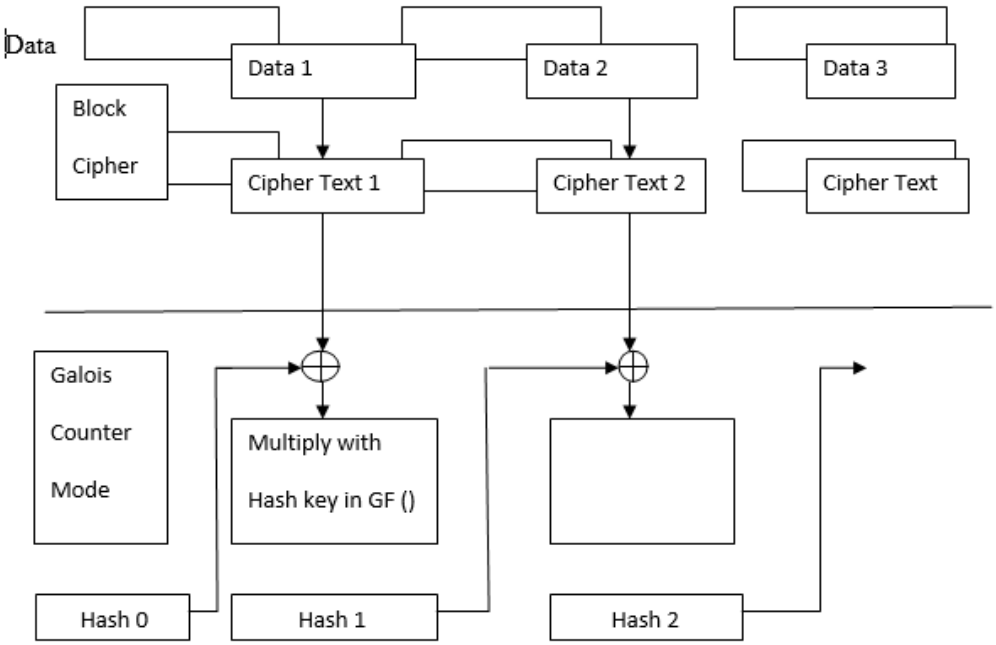

Figure 3.10 GCM generation for message digest – Galois Hash

AUTHENTICATED ENCRYPTION WITH ASSOCIATED DATA (AEAD) MODES

This mode provides integrity, confidentiality and authenticity of data in a cryptographic scheme. The Cipher operation in CTR, CFB, CBC, ECB and XTS provide different levels of data confidentiality but data authenticity and integrity of data. An recent Symantec report, Mobile users are putting their employers data, sharing logins and passwords with families (21%), and friends (18%), store personal and work information on line storage accounts (24%), and storing on line sensitive files (52%), the commonly exploited vulnerabilities are related to SSL and TLS protocol.

A Cisco report (CISCO 2014) indicates that cipher attacker gain access to name servers, hosting servers and data servers. Threat alerts are increasing every year by more than 10%. Most of the mobile malware target Android devices. The researcher (Bergman et al. 2013) observed that a data security is a concern in cloud computing, mobile computing, computer networks, mobile and web applications using rich multimedia systems is also a concern.

Ponemon Institute survey studies (Mehra, P. 2012) points that increased security threats resulted from increased use of smart mobile phones, increased use of cloud computing, increased sophistication of cyber attackers, and increasing volume of malware, absence of end to end connectivity security integration provisions.

Protocols IPSec Stack of Security

IPSec is an open standard for providing private secure communications over Internet Protocol (IP) protocols, by identifying various encryption and authentication algorithm and provide cryptographic keys required for services.

Security related problems of transfer of confidential data related to rich multimedia (RMM) data of video, audio and graphics over Internet are ensured by the design, development, and implementation is a data exchange protocol of communication, security set is a key aspect of dynamic VPN tunnels for user

security requirements, (Alsiherov, F., & Kim, T. 2010) and (Thulasimani, L., & Madheswaran, M. 2010).

VPN Technology and IPSec Stack of Protocols

The researcher (Alsiherov, F., & Kim, T. 2010) suggested the use of Virtual Private Network (VPN) technology and (Gepner, P., & Kowalik, M. F. 2006) proposed the IPSec Stack of security protocols which he consider as one of the efficient approaches to significantly reduce security concern in computer network for transmission of rich multimedia (RMM) data Video, audio and graphics, etc over public Internet.

Computer Security Institute published the report regarding the user's satisfaction rates with respect to computer network security and observed high satisfaction level for Firewalls provision, Encryption for data transmission, VPN, and one time password for Smart cards. The researchers (Trichina, E., & Korkishko, L. 2004), (Alsiherov, F., & Kim, T. 2010) suggested the key aspect of mobile (MVPN) design, development and implementation set of security, data exchange protocols, communication and dynamic VPN tunnels for mobility and security.

Protocols IPSec Stack of Security

IPSec is an open standard for providing private secure communications over Internet Protocol (IP) protocols, by identifying various encryption and authentication algorithms to be used and provide cryptographic keys required for services.

Security related problems of transfer of confidential data related to rich multimedia (RMM) data of video, audio and graphics over Internet are ensured by the design, development, and implementation is a data exchange protocol of communication, security set is a key aspect of dynamic VPN tunnels for user security requirements, (Alsiherov, F., & Kim, T. 2010) and (Thulasimani, L., & Madheswaran, M. 2010).

Modes of Operations of Encryption Algorithms

The researchers (Thulasimani, L., & Madheswaran, M. 2010), and (Alsiherov, F., & Kim, T. 2010) observed that the performance, efficiency and security in IPSec based Mobile VPN depend on the selected components of VPN tunnel such as:

a) Modes of Cipher Operations,
b) Authenticated encryption with associated data (AEAD) Mode.
c) Encryption algorithms, AES, or RC6 etc.
d) Authentication algorithm, HMAC (SHA-256)
e) Integrity algorithm, SHA_256 or SHA-512, etc.

THE EAX MODE OF CIPHER OPERATION

The researcher (Bellare, M., & Kohno, T. 2003) proposed EAX mode, which has 2-phase cryptography, in the first phase it checks the privacy of the message, and in second phase it checks the authenticity of the data. The EAX mode is based on CTR and OMAC (One key message authentication code). The security proof relies on a result about the security of a tweak able extension of OMAC in which an adversary can obtain a tag for the message of its choice and associated key-stream. This mode has following advantages:

1. Since it carries out encryption in first phase to ensure privacy and it checks the Authenticity in second phase, the invalid messages will be rejected in first phase itself.
2. It has on line capability to process streaming data on the fly; it is very much desired for Streaming RMM data and systems.
3. It can pre-process static headers.
4. It requires neither complex encodings nor aligned operations.

The researcher (Uskov, A. V. 2012) conducted the performance testing of different Block Cipher in EAX mode in IPSec based MVPN for RMM data on various testing platforms and observed that:

a) The AES-128 cipher in EAX mode has the best overall median in encryption performance on powerful Dell Laptop in single and in multi-processor modes of CPU operation, on Window and Linux OS in comparison to RC6 and RC5 Ciphers, for RMM test data of all sizes files.

b) The AES-128 cipher in EAX mode is almost 10 times better performance on Dell Laptop rather than on Generic Asus net-book with windows OS, and 6 times with Linux OS.

c) Performance of cipher in 2-phase EAX mode is lower than in 1-phase CTR mode as expected, since it uses both CTR mode and OMAC modes. However this mode provides Integrity, Confidentiality and data authenticity in this scheme.

For Mobile Uses requiring the highest security with data integrity and authenticity for transferring confidential data files through IPSec based MVPN AES-128 cipher in the EAX must be used. For Mobile User requiring higher level of performance but satisfactory level of security must go for AES-128 cipher in CTR mode.

CHAPTER 4

IMPLEMENTING SUBSTITUTION BOX OF AES

CHAPTER 4 – IMPLEMENTING SUBSTITUTION BOX OF AES

PRACTICAL IMPLEMENTATIONS OF S-BOX OF AES

S-Box is implemented normally by using look up tables (LUT) in which 256 predefined values of S-Box and the same numbers for Inverse S-Box are stored in a ROM, it offers a shorter critical depth, it is suitable for FPGA implementation in terms of gate count. In high speed pipelined designs unbreakable delay of LUT becomes drawback. The efficiency of AES hardware implementation in terms of speed, security, size and power consumption largely depend on its architecture Every attempt have been made by researchers to optimize one or more parameters for some specific application, either to reduce the chip area, power consumption or to increase efficiency, throughput, and security level. The different applications of society requirements demand different parameters with respect to size for mobile applications, high speed processing for quick response, (Sasaki, Y. 2011), by researcher (Uddin, M., & Rahman, A. A. 2010) architecture in VLSI was proposed for single FPGA chip pipelined design (Stevens, K., & Mohamed, O. A. 2005), for high throughput with fully pipelined FPGA implementation. (Kim, J., Hong, S., Sung, J., Lee, S., Lim, J., & Sung, S, 2003). Design can be based on logic synthesis using Truth table **or** direct implementation of Algebraic Normal Form (ANF) expression for each column.

S-Box transformation in AES Implementation is the nonlinear transformation and it provides confusion part in encryption of data processing and contributes significant part in achieving high security. CFA based optimization is used for reducing area for FPGA or VLSI designs for compact mobile applications, the data security is ensured by adopting different masking techniques.

Algorithmic and CFA architectural optimization can be achieved in basic representations by elimination of redundant common factors in the inverter, appropriate choice of the field polynomials is required, and minimize the arithmetic complexity by merger of some multipliers with some sub-operations.

The sum of the upper and lower halves of each factor can be shared between two or more sub-field multipliers, which have the same input factor, one XOR addition is saved in 2-bit factor shared by two GF (2^2). Five XOR s are saved in 4bit factor shared by two GF (2^4) multipliers. Area saving is achieved on combining GF 2^2) multiplier with a scalar in a GF (2^4) multiplier, their results a saving of three XORs in total gates and one XOR in critical path. On combining the sum of upper and lower halves of the inputs of multiplier, common factors with GF (2^4) and square scalar there will be reduction of two XORs inverter. We can save around 30 XORS gates in the total gates and 3 XORS gates in the critical depth.

The common and straight forward implementation of the S-Box for SubByte transformation is by using pre-computed values stored in PROM based on Lookup table for encryption of data and the InvSubByte transformation by using another Lookup table of inverse S-Box for decryption to obtain decipher data in the receiver output. The different applications of society requirements demand different parameters with respect to high throughput rate for server application, compact in size for mobile applications, high speed processing for quick response and high security level by long security key size. S-Box transformation in AES Implementation is the nonlinear transformation and it provides confusion part in encryption of data processing and contributes significant part in achieving high security.

IMPLEMENTATION OF S-BOX USING COMBINATIONAL LOGIC CIRCUITS

Pipelining can be applied to S-box implementation for high throughput and compactness as compared to ROM based lookup table implementation, which has fixed access time, since ROMs have a fixed access time for its write. The SubByte transformation is computed by taking the multiplicative inverse in GF (2^8) followed by an affine transformation (AT). InvSubByte is calculated by applying inverse affine transformation ($AT^{-1)}$ before computing multiplicative inversion in GF (2^8).

SubByte: » Multiplicative Inversion in GF (2^8) » Affine Transformation

InvSubByte: » Inverse Affine Transformation » Multiplicative Inversion in GF (2^8)
The Affine Transformation AT (a):

$$AT(a) = \begin{pmatrix} 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \\ 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \end{pmatrix} \times \begin{pmatrix} a7 \\ a6 \\ a5 \\ a4 \\ a3 \\ a2 \\ a1 \\ a0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Figure 4.1 Affine Transformations AT (a)

Inverse Transformation AT^{-1} (a):

$$AT^{-1}(a) = \begin{pmatrix} 0\,1\,0\,1\,0\,0\,1\,0 \\ 0\,0\,1\,0\,1\,0\,0\,1 \\ 1\,0\,0\,1\,0\,1\,0\,0 \\ 0\,1\,0\,0\,1\,0\,1\,0 \\ 0\,0\,1\,0\,0\,1\,0\,1 \\ 1\,0\,0\,1\,0\,0\,1\,0 \\ 0\,1\,0\,0\,1\,0\,0\,1 \\ 1\,0\,1\,0\,0\,1\,0\,0 \end{pmatrix} \begin{pmatrix} a7 \\ a6 \\ a5 \\ a4 \\ a3 \\ a2 \\ a1 \\ a0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Figure 4.2 Inverse Transformation AT^{-1} (a)

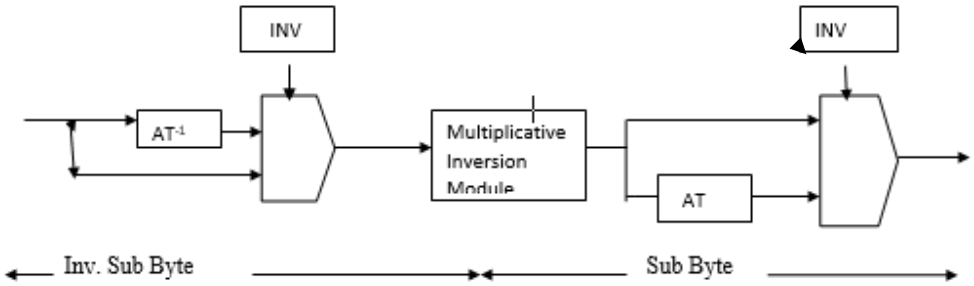

Figure 4.3 Combined InvSubByte and SubByte with common Multiplicative Inversion Module

The AT and AT^{-1} are the Affine Transformation and its inverse while the vector is the multiplicative inverse of the input byte from the state array. In hardware architecture as shown in Figure 5.3, switching between SubByte and InvSubByte is changing the value of INV signal. INV is set to 0 for SubByte while 1 is set when InvSubByte operation is desired.

The S. box has the advantage of small area and may be pipelined for increased performance in clock frequency.

S-BOX ARCHITECTURE

The efficiency of the hardware implementation of the AES algorithm in terms of security power consumption, area and speed depends on the structure of S-box since power consumption and resources on implementation schemes. The Resources limited systems such as wireless sensor networks and radio Frequency and sensor.

The multiplicative inverse computation will be done, and will then the Affine transformation will follow for construction of S-Box for the SubByte. The operation of Inverse Affine transformation will be followed by multiplicative Inversion module for the InvSubByte , as shown in Figure 4.3.

The mapping of Multiplicative Inverse of S-Box is to be done first, from Galois finite field GF (2^8) by Composite Field Arithmetic to $GF(((2^4)^2)$ and generate expressions for all sub-operations over $GF(2^4)$, then reduce redundant resources

in sub-operations and isomorphic mapping matrices. On optimization, it has been found (Standaert, F. X., Peeters, E., & Quisquater, J. J. 2005) that normal basis better results obtained in respect of smaller area and shorter critical path than polynomial basis model.

MULTIPLICATIVE INVERSION MODULE IMPLEMENTATION

A byte representing a GF (2^8) element can be viewed as coefficients to each power term in the GF (2^8) polynomial. The data byte $\{10101011\}_2$ is representing the polynomial q7 + q5 + q3 + q + 1 in GF (2^8). Any arbitrary polynomial can be represented as b x + c, given an irreducible polynomial of x^2 + Ax + B.

The element in GF (2^8) when represented as b x +c, b is the most significant nibble while c is the least significant nibble, then the multiplicative inverse can be computed using the under mentioned equation .

$$(b\ x + c)^{-1} = b\ (b2\ B + b\ b\ A + c^2)^{-1}\ x + (c + b\ A)\ (b2\ B + b\ c\ A + c^2)^{-1} \quad (5.1)$$

From (Akkar, M. L., & Giraud, C. 2001), the irreducible polynomial that was selected was x2 +x + λ. Since A = 1 and B = λ, then the equation could be simplified to the form as shown below.

$$(b\ x + c)^{-1} = b\ (b^2\ \lambda + c\ (b + c))^{-1}\ x + (c + b)\ (b^2\ \lambda + c\ (b + c))^{-1} \quad (5.2)$$

There are addition, multiply, squiring and multiplication inversion in GF (2^4) operations in Galois Field. For computing multiplicative inverse, operators of the equation can be converted in individual blocks for constructing the circuit of multiplicative inverse.

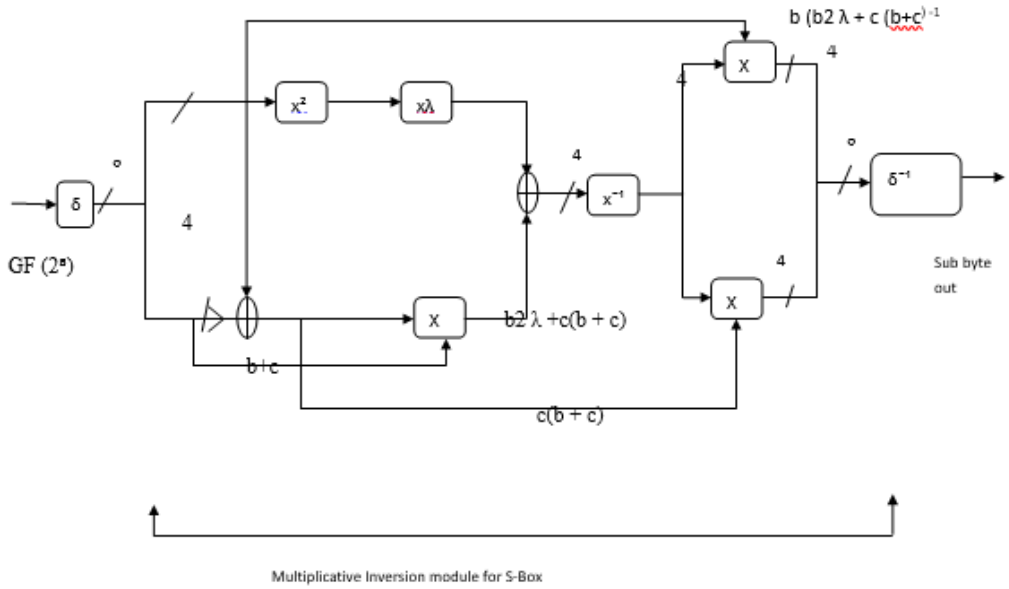

Figure 4.4 A conventional S- Box architecture in composite field7

DERIVATION OF MULTIPLICATIVE INVERSE IN S-BOX ALGORITHM USING SUBFIELDS IN CFA

While performing S-box function during encryption of data on blocks of bytes in GF (28) with field polynomial g(x) = x8 +x4 +x3 +x +1, lot of resources are consumed in finding a multiplicative inverse and affine transformation operation, Rijmen suggested higher order field conversion to subfields to reduce complexity, simply calculations.

First map elements of field P to a composite field Q using isomorphism function;

$$q = f(p) = \delta \times p;$$

Second compute the multiplicative inverse over Q;

$$X = q\text{-}1 \text{ (except if } q = 0, \text{ then } x = 0);$$

Third remap the computation results to P, using the inverse isomorphism function;

$$P = f\,`(x) = \delta\text{-}1 \times x \ .$$

In mapping a computation from one field to another field to find most efficient implementation scheme, the Galois Field GF (28) is mapped to GF (((22)2)2) it requires 3 stages of isomorphism and field polynomials. The binary 8X8 matrix-vector product in isomorphic mapping can be expressed as eight bit-level equations. CSE would be useful in extracting common factors from these bit-level equations to reduce the area cost and the critical path.

IMPLEMENTATION OF S-BOX USING COMPOSITE FIELD ARITHMETIC (CFA) ARCHITECTURE

The circuit can be coded in hardware description language i.e. VHDL or Verilog manually. Then ANF representation with fine-grained pipeline registers can be inserted to check the feasibility and throughput rates, multiplication sub-operation can be put into two parts for fine-grained pipelining. The structure of the S-Box is given in Figure 5.5. S-Box area can be minimized by clubbing the inverse isomorphic mapping and Affine Transformation which will reduce the slices required for S-Box Implementation as shown in Figure 5.6 and . A 2-layer pipeline is used to break the logic delay in the attempt to achieve higher clock frequency.

The common sub-expressions are identified for its elimination and replace them with a variable to reduce the redundant resources in S-box's multiplicative inversion circuit of GF (24), so that gate count are reduced significantly in S-Box circuit design. The highest occurring variable pattern frequency is monitored for its elimination for S-Box optimization. The elimination patterns are to be generated to identify the occurrence frequency of variables of N-terms patterns in computation equation, systematic process of elimination of highest frequency N-term and replacement with new variable is to be carried out for circuit optimization. Elimination of N-terms is continued until no occurring N-terms are observed. Polynomial basis and Normal basis structures are designed to optimize for low delay and small area. The researcher Zhang achieved the

best area-delay product with Normal basis structure than Polynomial basis structure.

S-Box based on Composite Field Arithmetic (CFA) Architecture for high throughput has been proposed (Uskov, A., Byerly, A., & Heinemann, C. 2016), a sequence of algorithm and architectural optimization for each composite field construction is to be verified. There are eight possible isomorphic mappings, common sub-expression elimination algorithm may be developed to choose mapping with minimal implementation area cost. Through algebraic normal form and fine-grained pipelined designing architecture, we can achieve a 3.0 Gaps' throughput in FPGA chip. The smallest CFA based S-box (Canright, D. 2005), however a short critical path is also desired in chips architecture, deep sub -pipelining for increased performance is also desired. S-box of AES with shortest critical path was proposed (Standaert, F. X., Peeters, E., & Quisquater, J. J. 2005), but it required a large area compared (Canright, D., & Batina, L. 2008) for S-Box. (Uskov, A., Byerly, A., & Heinemann, C. 2016) proposed the optimal s-box with the shortest possible critical path with minimum silicon area. Mapping of field, basis representation, mapping of isomorphic and field polynomials are four focus of point for CFA optimizations. All the eight possible isomorphic mappings must be examined. Wong worked on Sub sharing optimization in matrix multiplication, common sub expression algorithm to reduce area in isomorphic mappings. Fine-grained pipelining to the GF (24) multiplier was applied by Wong to improve the performance of CFA based S-box by using AND, and XOR operations in algebraic normal form (ANF) representation, to achieve minimum area cost and highest throughput.

DERIVATION OF MULTIPLICATIVE INVERSE IN S-BOX ALGORITHM USING SUBFIELDS IN CFA

While performing S-box function during encryption of data on blocks of bytes in GF (28) with field polynomial $g(x) = x8 + x4 + x3 + x + 1$, lot of resources are consumed in finding a multiplicative inverse and affine transformation operation, Rijmen suggested higher order field conversion to subfields to reduce complexity, simply calculations.

First map elements of field P to a composite field Q using isomorphism function;

$$q = f(p) = \delta \times p;$$

Second compute the multiplicative inverse over Q;

$$X = q\text{-}1 \text{ (except if } q = 0, \text{ then } x = 0);$$

Third remap the computation results to P, using the inverse isomorphism function;

$$P = f`(x) = \delta\text{-}1 \; X \; x.$$

In mapping a computation from one field to another field to find most efficient implementation scheme, the Galois Field GF (28) is mapped to GF (((22)2)2) it requires 3 stages of isomorphism and field polynomials as follows.

$$a(y) = y2 + ay + v \text{ (isomorphism for GF(28) / GF(24))} \qquad (1)$$

$$b(z) = z2 + T\,z + N \text{ (isomorphism for GF(24) / GF (22)} \qquad (2)$$

$$c(w) = w2 + w + 1 \text{ (isomorphism for GF (22) / GF (2))} \qquad (3)$$

For equation (1) and (2) above we have to determine all the possible coefficients of v, t, n, and T in both normal and polynomial bases. Polynomial basis representation has been used (Standaert, F. X., Peeters, E., & Quisquater, J. J. 2005), normal basis representation was used (Canright, D. 2005).

$w2 + w + 1 = 0$ are the irreducible polynomial over GF (2).

OPTIMUM ISOMORPHIC MAPPING WITH COMMON SUB EXPRESSION ELIMINATION

The binary 8X8 matrix-vector product in isomorphic mapping can be expressed as eight bit- level equations. CSE would be useful in extracting common factors from these bit-level equations to reduce the area cost and the critical path.

OPTIMIZATION OF CFA ARCHITECTURES

We manually coded the circuit using a hardware description language for all of the three proposed CFA S-boxes. Employ ANF representation along with a strategic fine-grained pipeline registers insertion, in an attempt to validate the feasibility of compact CFA AES S-boxes.

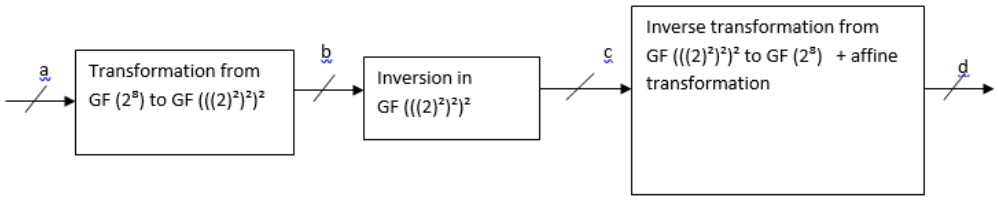

Figure 4.5 Structure of the S- Box implementation

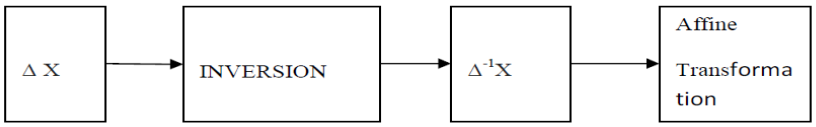

Figure 4.6 (a) Implementation of S-Box of AES

$$\Delta = \begin{pmatrix} 1\,1\,0\,0\,0\,0\,1\,0 \\ 0\,1\,0\,0\,1\,0\,1\,0 \\ 0\,1\,1\,1\,1\,0\,0\,1 \\ 0\,1\,1\,0\,0\,0\,1\,1 \\ 0\,1\,1\,1\,0\,1\,0\,1 \\ 0\,0\,1\,1\,0\,1\,0\,1 \\ 0\,1\,1\,1\,1\,0\,1\,1 \\ 0\,0\,0\,0\,0\,1\,0\,1 \end{pmatrix}^{2} \qquad \delta^{-1} = \begin{pmatrix} 1\,0\,1\,0\,1\,1\,1\,0 \\ 0\,0\,0\,0\,1\,1\,0\,0 \\ 0\,1\,1\,1\,1\,0\,0\,1 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,1\,1\,0\,1\,1\,1\,0 \\ 0\,1\,0\,0\,0\,1\,1\,0 \\ 0\,0\,1\,0\,0\,0\,1\,0 \\ 0\,1\,0\,0\,0\,1\,1\,1 \end{pmatrix}^{2}$$

Figure 4.6 (b) Matrix and Inverse Matrix

ISOMORPHIC MAPPING AND INVERSE ISOMORPHIC MAPPING

The multiplicative inverse computation will be done by decomposing complex GF (28) to lower order fields of GF(21), GF (22), and GF((22)2), the following irreducible polynomials are used. (Akkar, M. L., & Giraud, C. 2001)

$$GF(22) \text{ » } GF(2) : x2 + x + 1$$

$$GF((22)2) \text{ » } GF(22) : x\,2 + x + \varphi \qquad (5.3)$$

$$GF(((22)2)2) \text{ » } GF((22)2) : x2 + x + \lambda$$

Where $\varphi = \{10\}2$ and $\lambda = \{1100\}2$

Computation of the multiplicative inverse in composite fields of GF (28) element cannot be directly applied, first element has to be mapped to its composite field representation via an isomorphic function, δ. The result of multiplicative inversion will be mapped back from composite field to its equivalent in GF (28) i.e. the inverse isomorphic function δ-1. Let q be the element in GF (28), then the isomorphic mappings and its inverse can be written as δ *q and δ-1 *q, which is a case of matrix multiplication as in Figure Where q7 is the most significant bit and q0 is the least significant bit.

$$\Delta \times q = \begin{pmatrix} 1\,0\,1\,0\,0\,0\,0\,0 \\ 1\,1\,0\,1\,1\,1\,1\,0 \\ 1\,0\,1\,0\,1\,1\,0\,0 \\ 1\,0\,1\,0\,1\,1\,1\,0 \\ 1\,1\,0\,0\,0\,1\,1\,0 \\ 1\,0\,0\,1\,1\,1\,1\,0 \\ 0\,1\,0\,1\,0\,0\,1\,0 \\ 0\,1\,0\,0\,0\,0\,1\,1 \end{pmatrix} \times \begin{pmatrix} q7 \\ q6 \\ q5 \\ q4 \\ q3 \\ q2 \\ q1 \\ q0 \end{pmatrix}$$

Figure 4.7 (a) Matrix Multiplication

$$\Delta^{-1} \times q = \begin{pmatrix} 1\,1\,1\,0\,0\,0\,1\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0 \\ 0\,1\,1\,0\,0\,0\,1\,0 \\ 0\,1\,1\,1\,0\,1\,1\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 1\,0\,0\,1\,1\,1\,1\,0 \\ 0\,0\,1\,1\,0\,0\,0\,0 \\ 0\,1\,1\,1\,0\,1\,0\,1 \end{pmatrix} \times \begin{pmatrix} q7 \\ q6 \\ q5 \\ q4 \\ q3 \\ q2 \\ q1 \\ q0 \end{pmatrix}$$

Figure 4.7(b) Matrixes Multiplicative Inversion

Logical XOR operation can be derived from above matrix multiplication, as shown below:

$$\Delta \times q = \begin{pmatrix} q7 \oplus q5 \\ q7 \oplus q6 \oplus q4 \oplus q3 \oplus q2 \oplus q1 \\ q7 \oplus q5 \oplus q3 \oplus q2 \\ q7 \oplus q5 \oplus q3 \oplus q2 \oplus q1 \\ q7 \oplus q6 \oplus q2 \oplus q1 \\ q7 \oplus q4 \oplus q3 \oplus q2 \oplus q1 \\ q6 \oplus q4 \oplus q1 \\ q6 \oplus q4 \oplus q0 \end{pmatrix}$$

Figure 4.8 Logical XOR Operations

$$\Delta^{-1} \times q = \begin{pmatrix} q7 \oplus q6 \oplus q5 \oplus q1 \\ q6 \oplus q2 \\ q6 \oplus q5 \oplus q1 \\ q6 \oplus q5 \oplus q4 \oplus q2 \oplus q1 \\ q5 \oplus q4 \oplus q3 \oplus q2 \oplus q1 \\ q7 \oplus q4 \oplus q3 \oplus q2 \oplus q1 \\ q5 \oplus q4 \\ q6 \oplus q5 \oplus q4 \oplus q2 \oplus q0 \end{pmatrix}$$

Figure 4.9 Inverses Isomorphic Mapping

The matrix multiplication can be translated to logical XOR operation. The logical form of the above matrices is as given below:

5.4.1 Composite Field Arithmetic operations

5.4.2 Addition in GF (2^4)

5.4.3 Squiring in GF (2^4)

5.4.4 Multiplication with constant, λ

5.4.5 GF (2^4) Multiplication

5.4.6 GF (2^2) Multiplication

5.4.7 Multiplication with constant φ

5.4.8 Multiplicative Inversion in GF (2^4)

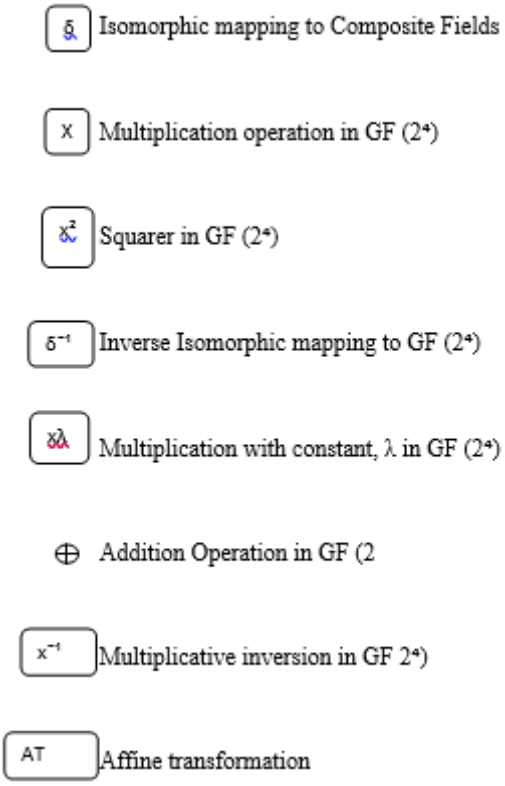

Figure 4.10 Meaning of symbols used in Mapping

FPGA IMPLEMENTATION OF CFA VERSION OF S-BOX

S-Box area can be reduced by merging the inverse isomorphic mapping with the Affine Transformation. The implementation of δ-1 and Affine Trans formation module can be combined to reduce the slices occupied by the S-Box. A 2-layer pipeline is used to break the logic delay in the attempt to achieve higher clock frequency.

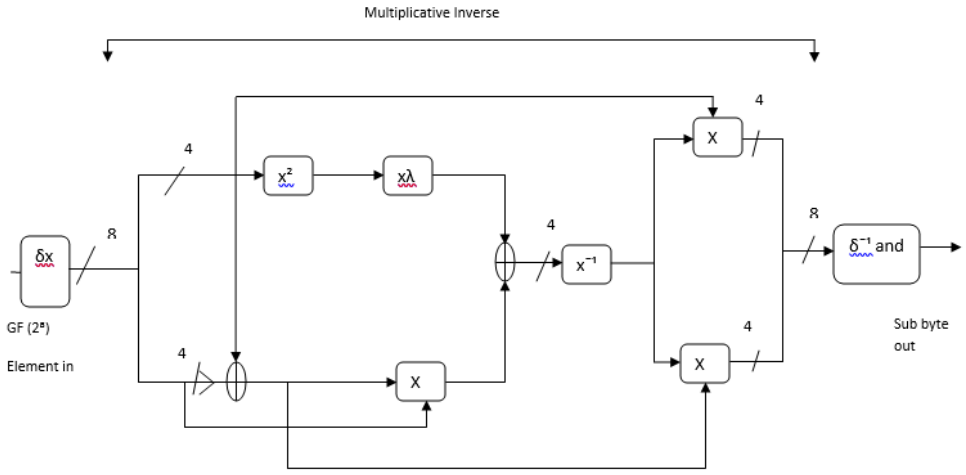

Figure 4.11 A conventional S- Box architecture in composite field 7

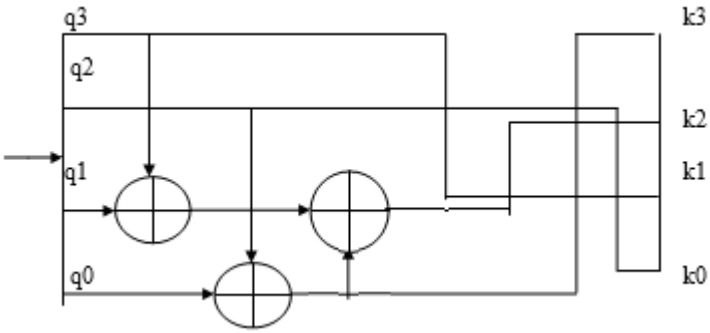

Figure 4.12 Hardware diagram for multiplication withy constant λ

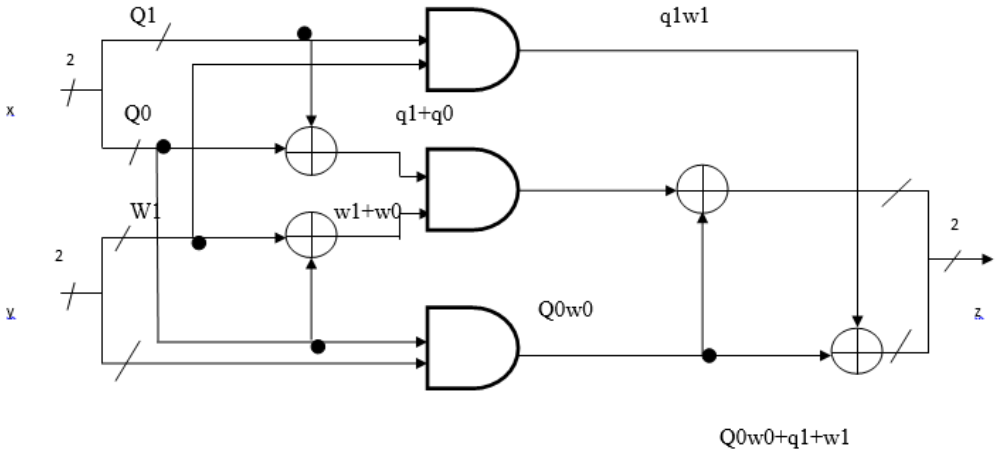

Figure 4.13 Hardware Implementation of multiplication in GF (2)

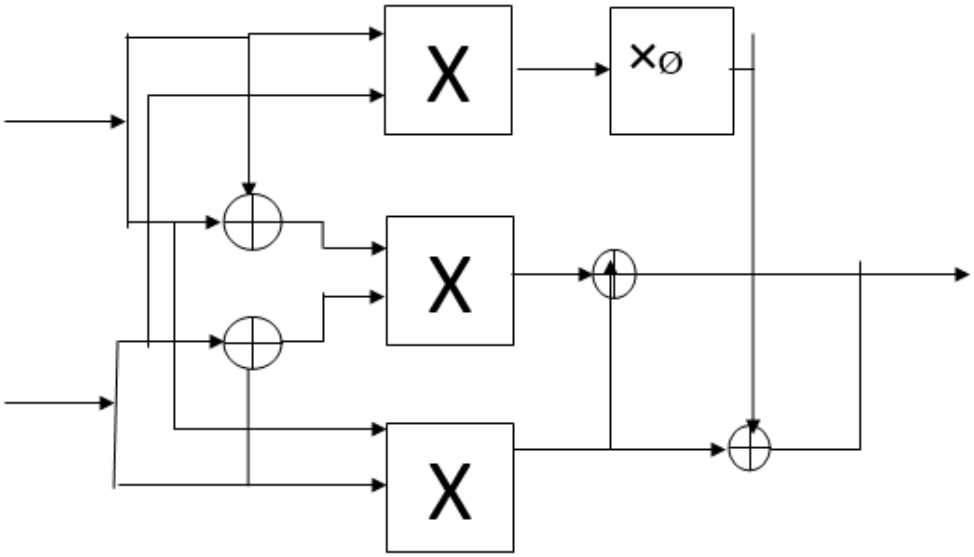

Figure 4.14 Hardware Implementation of multiplication in GF (2^4)

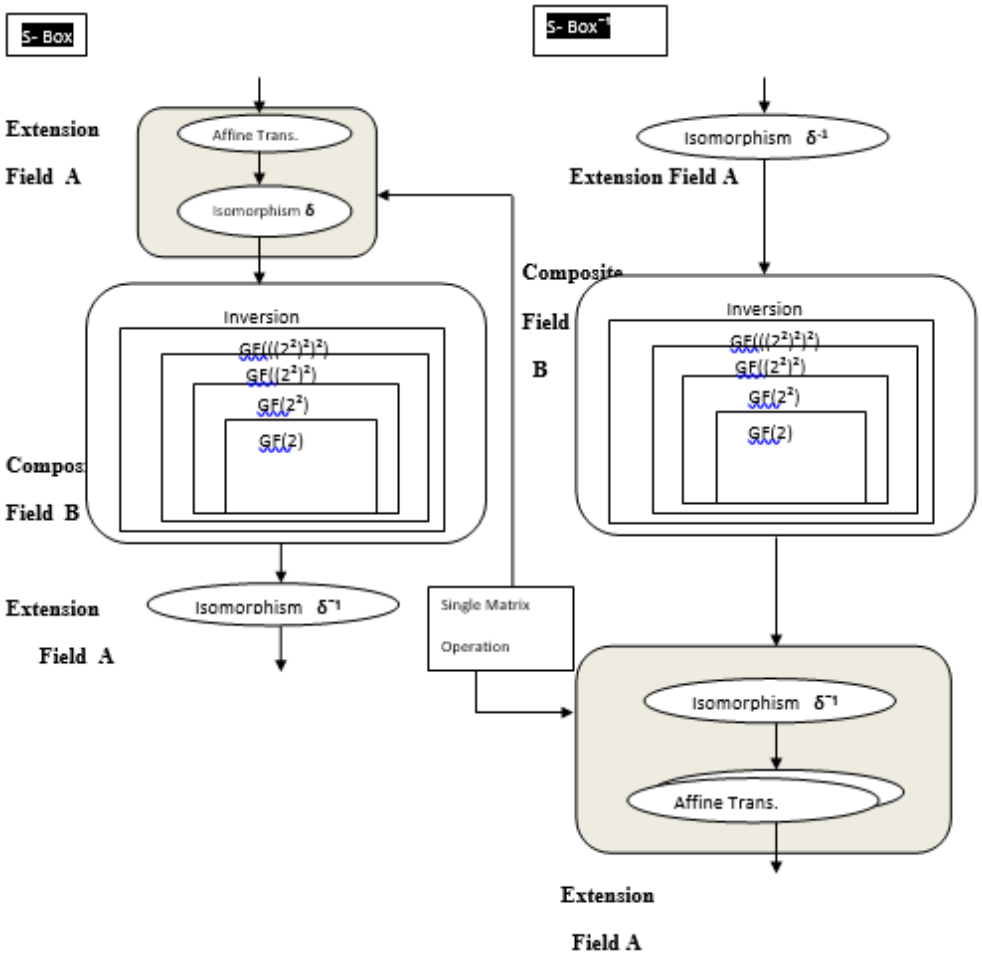

Figure 4.15 Computation Sequence of S-Box Implementation

HIGH PERFORMANCE ARCHITECTURE OF AES

It uses pipelined structure to increase the throughput of Mix Column and CFA S-box. The composite field is constructed not by applying a single degree 8 extension to GF (2), but by applying multiple extensions of smaller degrees. In order to reduce the cost it is better to be efficient to construct the composite field using repeated degree -2 extensions under polynomial basis using these irreducible polynomials.

GF (24) : x4 +x +1

GF ((24)2) : x2 +x + w14

Where w14 : {1001}2

OPTIMIZED CFA S-BOXES OF AES

The Rijmen researcher proposed the method of calculating multiplicative inverse of higher order field element , by mapping GF (28) element into its subfield in order to generate less complex multiplicative inverse. The manipulation in subfields is easier in comparison to higher order GF field, iteratively from lower order subfields composite field architecture may be build.

a) Map first all elements of field P into a composite field Q by isomorphism function; q = f (p) = δ x p;

b) Compute the multiplicative inverse of Q: x = q-1 (except if q=0, then x= 0);

c) Using inverse isomorphism function, remap computation to P; p = f' (x) =q -1 x x.

For finding most efficient implementation, isomorphism can be used to map a computation from one field to another.

Mapping Galois field from higher order GF (28) into lower order GF (((22)2)2) requires 3 field polynomials and isomorphism.

$$k(y) = y2 + x\,y + v \quad \text{(isomorphism GF (28) / (GF (24))} \qquad (1)$$

$$l(z) = z2 + T\,z + N \quad \text{(isomorphism GF (24) / GF (22))} \qquad (2)$$

$$m(w) = w2 + w + 1 \quad \text{(isomorphism GF (22) / GF (2).} \qquad (3)$$

For constructing compact CFA combinational circuitry one has to select proper field of mapping, field of polynomials, basis representation and isomorphic mapping. The minimum area of occupancy has to be preserved and work for finding shortest critical path for the circuit. Sub sharing in binary matrix multiplication in AES algorithm may be used for optimization purpose. The area minimization in isomorphism mapping may be applied by using common sub-expression elimination algorithm. CFA S- boxes of AES are designed as direct computation modules consisting of, Gate and XOR Gate operations, in this way minimum area cost, and higher data throughput can be achieved.

Proper selection of the coefficients of field polynomials is required for minimal arithmetic complexity. The elimination of redundant common factor can be done in the inverter circuit. and some multipliers along with sub operations can also be merged. Some of the sub field's multipliers, which have same input factor, may be shared for higher and lower halves. The 2 bit factor shared by two GF (22) multiplier saves one XOR addition. The 4 bit factor shared by two GF (24) multipliers saves 5 XORs. Strategic fine-grained register insertion and ANF representation together will help in simplifying CFA architectures.

HIGH THROUGHPUT OPTIMIZED CFA BASED COMPACT S-BOXES

A design for optimizing composite field architecture for achieving high throughput for S-Boxes is proposed (Uskov, A., Byerly, A., & Heinemann, C. 2016). There are eight possible isomorphic mappings, a common sub-expression elimination algorithm may be developed to choose mapping with minimal implementation area cost. Through algebraic normal form and fine-grained pipelined designing architecture, we can achieve a 3.0 Gbps throughput in FPGA chip. The smallest CFA based S-box was proposed (Canright, D. 2005), however a short critical path is also desired in chips architecture, deep

sub -pipelining for increased performance is also desired. S-box of AES with shortest critical path was proposed (Standaert, F. X., Peeters, E., & Quisquater, J. J. 2005), but it required a large area compared to Canright's S-Box. (Rizk, M. R. M., & Morsy, M. 2007) proposed the optimal s-box with the shortest possible critical path with minimum silicon area. Mapping of field, basis representation, mapping of isomorphic and field polynomials are four focus of point for CFA optimizations. All the eight possible isomorphic mappings must be examined.

Fine-grained pipelining to the GF (24) multiplier was applied by Wong to improve the performance of CFA based S-box by using AND, and XOR operations in algebraic normal form (ANF) representation, to achieve minimum area cost by applying common sub-expression elimination (CSE) algorithm to reduce to reduce area in isomorphic mapping.

OPTIMIZATION OF CFA ARCHITECTURE

Algorithmic and architectural optimization can be achieved in basis representations by elimination of redundant common factors in the inverter, precise selection of the field polynomials chosen, and minimize arithmetic complexity by merger of some multipliers with some sub-operations. The sum of the higher and lower halves of each factor can be shared between two or more sub-field multipliers, which have the same input factor, one XOR addition is saved in 2-bit factor shared by two GF (22). Five XOR s are saved in 4bit factor shared by two GF (24) multipliers. Area saving is achieved on combining GF 22) multiplier with a scalar in a GF (24) multiplier, their results a saving of 3 XORs in total gates and one XOR in critical path.

On combining the sum of higher and lower halves of the inputs of multiplier, common factors with GF (24) and square scalar there will be reduction of 2 XORs inverter. We can save around 30 XORS gates in the total gates and 3XORS gates in the critical path.

HARDWARE IMPLEMENTATION OF CFA S-BOXES

The circuit can be coded in hardware description language i.e. VHDL or Verilog manually. Then ANF representation with fine-grained pipeline registers can be

inserted to check the feasibility and throughput rates, multiplication sub-operation can be put into two parts for fine-grained pipelining.

S-Box area can be reduced by merging the inverse isomorphic mapping with the Affine Transformation. The implementation of δ-1 and Affine Trans formation module can be combined to reduce the slices occupied by the S-Box. A 2-layer pipeline is used to break the logic delay in the attempt to achieve higher clock frequency.

MVP-CSE ALGORITHM FOR COMPACT S-BOX

CFA Architecture is used to minimize the silicon area and reduce the critical path. The researcher (Standaert, F. X., Peeters, E., & Quisquater, J. J. 2005) proposed Multi- Variable Pattern Common Sub-expression Elimination (MVP-CSE) algorithm to minimize or eliminate the redundant resources in GF (24) Multiplicative Inverter in Normal basis S-box, obtain low value of critical path compared with the polynomial basis and isomorphism mapping functions. Normal basis S-box has the low value of delay and requires less area. The hardware implementation efficiency in terms of speed, area, power consumption and security depends on architecture selected for the S-box. Area optimization of AES hardware is highly desired in resources-limited systems.

Optimized for area S-box designs based on CFA architecture was proposed by researcher (Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., & Shamir, A. 2010), and by (Canright, D. 2005) presented normal basis smallest S-box but it had long critical path. Zhang based on polynomial presented shortest critical path S-box, however it requires large area. In order to obtain low critical path and small gate count Squarer module and constant multiplier module are merged in CFA implementation. CFA Architecture is used to reduce the area consumption and shorten the critical path. Multi- Variable Pattern Common Sub-expression Elimination (MVP-CSE) algorithm proposed by (Standaert, F. X., Peeters, E., & Quisquater, J. J. 2005). Normal basis S-box is further optimized to reduce redundant resources of multiplicative inversion in GF (24) module and a shorter critical path compared with the polynomial basis and isomorphism mapping functions. Normal basis S-box has the smallest area

delay. The hardware implementation efficiency in terms of speed, area, power consumption and security depends on implementation of the S-box. Area optimization of AES hardware is highly desired in resources-limited systems.

Small area S-box designs using CFA architecture were proposed in (Alsiherov, F., & Kim, T. 2010), (Anand, A., & Patel, B. 2012), (Aoki, K., & Sasaki, Y. 2009) and (Bellare, M., & Kohno, T. 2003). (Calomel.org 2015) presented normal basis smallest S-box but it had long critical path. Based on polynomial presented shortest critical path S-box, however it requires large area. Squarer module and the constant multiplier module in GF (24) are merged into a single one using CFA, to reduce the gate counts significantly, critical path also shortened significantly.

Composite Field Implementation of S-box: S-box is calculated using Eq. (1)

$$FT = M((XT) - 1) + VT \quad (1)$$

Where X is the input state matrix, M is an 8X8 constant matrix, defined as

M = [0x8E, 0xC7, 0xE3, 0xE3, oxF1, oxF8, 0x7C, 0x3E, 0x1F]. V is an 8-bit constant vector, defined as V = [0x63]. The row vectors of both M and V are represented in the hexadecimal format. The process of affine transformation is defined as follows: X-1 is multi plied by M first, then combined with a constant vector V in the affine transformation. The study focuses on approaches to deduce and simplify the multiplicative inversion in GF (28).

Based on the CFA technique, the S-box can also be calculated by using (2). Two parts are included in deriving the multiplicative inversion in GF (28). One is the derivation of an inversion module in the composite field GF (((22)2)2), and the other is the calculation of mapping matrices δ and δ-1 in isomorphism mapping functions. The mapping matrices δ and δ-1 are the linear transformations between the finite field GF (28) and the composite field GF (((22)2)2).

$$FT = M(\delta\text{-}1(\delta XT) - 1) + VT \quad (2)$$

The inversion module structure is derived from irreducible polynomial coefficients of composite fields

GF ((24)2), GF ((22)2), and GF (22), which are as follows:

GF (((22)2)2) : f (y) = y2 + τ y + v

GF ((22)2) : f (z) = z2 + T z + N

GF (22) : f (w) = w2 + w +1

Where τ = (000)4, T = (01)2, and v1 + 91100)4, N1 = (10)2 in Polynomial basis S-box.

δ is the mapping matrix from the finite field to the composite field , and δ-1 is the inverse of δ . The relationship between δ and δ-1 is δ x δ-1 = E and E is a unit matrix. The mapping matrix δ-1 can be combined with the affine matrix M to simplify the circuit structure of the S-box.

The structures of the inversion module and the value of the mapping matrix can be derived based on the irreducible polynomial coefficients of the composite field

GF ((22)2) and GF ((24)2). The irreducible polynomial of the composite field GF ((24)2), GF ((22)2) and GF (22) operations are denoted as follows:

GF (((22)2)2) : f (y) = y2 + τ y +v

GF ((22)2) : f (z) = z2 + T z +N

GF (22) : f (w) = w2 +w +1

Where τ = (000)4, T = (01)2, and v1 + 91100)4, N1 = (10)2 in Polynomial basis S-box.

CFA OPERATION FOR S-BOX OPTIMIZATION

Optimize the modules of the xv, a2, and the a-1 that are shown in Figure 1(b) and Figure1(c) of the S-box. Each module's implementation is derived by using the CFA technique to reduce the area and to increase speed.

If the common sub-expressions are present more than once then these are to be identified and represented as single variable to reduce the gate count in the S-Box. The patterns with some variables and highest occurring frequency are identified to eliminate at each of the iteration. The randomly candidate pattern is selected to eliminate greedy algorithm to check all possible patterns to identify the best set of patterns with minimal area. In order to reduce the gate count needed by optimized module the elimination patterns are to be generated. Occurrence frequency of N-term patterns in the equation is to be computed. A list to be generated of N-term patterns with the highest frequency. Identify the pattern with highest frequency for elimination systematically. The selected pattern is replaced with a new variable. The equation formed by selected patterns shall be further optimized. Next iteration shall be tried until no recurring N-term patterns observed. Compute the gate count needed by optimized module and observe the improvements achieved.

Both of the S-Box structures a polynomial basis and normal basis have been studied to reduce the area-delay product by combining xv and a2 modules. The technique suggested by Zhang is efficient in saving the resources and reducing the critical path also, normal basis structure achieves the best area – delay product as compared to polynomial basis.

ANOTHER HIGH PERFORMANCE ARCHITECTURE OF AES

It uses pipelined structure to increase the throughput of Mix Column and CFA S-box. Applying multiple extensions of smaller degrees in place of a single degree 8 extension to GF (2) is efficient while constructing the composite field. In order to reduce the cost it is better to be efficient to construct the composite field using repeated degree -2 extensions under polynomial basis using these irreducible polynomials.

GF (24) : x4 +x +1

GF ((24)2) : x2 +x + w14

Where w14 : {1001}2

For any Composite fields GF ((2m) n), computing the multiplicative inverses can be done as a combination of operations over the sub-fields GF (2n), using the following equation:

P -1 = (P r).P r-1

Where r = 2nm -1) / (2m-1)

For AES (n =2, m =22),

Therefore p -1 = (p -17. p 16)

The computation of P-1 is obtained by multiplying P 16 with P -17 over GF (((22)2)2) Because P17 is always an element of GF (((22)2)2), calculating upper 4 bits of P17 are not needed being always zero and hence saving. The value of (P17)-1 is computed recursively over GF (((22)2)2). The circuit gates of the three GF (((22)2)2) should be shared in order to minimize gate counts. The P17 element of GF (((22)2)2) is used to compute P-1 in order to reduce circuit resources than conventional multiplication over GF (28). The inverter and multipliers over GF (((22)2)2) and GF (22) sub fields are also small. In decryption process, first inverse affine transformation is taken first, then convert elements from GF(28) into two elements of GF ((2)4)2 and then inverter is used to obtain the inverted output.

The high performance architectures have been suggested for VLSI, ASIC implementation by (Talwar, Y., & Veni Madhavan, C. E. 2005), (Schramm, K., & Paar, C. 2006) VLSI chip, and (Uskov, A. 2014) for S-Box implementation. Every researcher optimizes his design for some specific application for higher throughput, smaller size for compact module, higher security level.

A Compact and Optimized S-Box proposed by researcher (Rais, M. H., & Al Mijalli, M. H. 2012) for Implementation of S-Box, the look-up table method requires two different circuits, an S-Box for encryption and inverse S-Box for decryption. A large amount of hardware around 1400 gates is required for each one byte S-Box based on the look-up table method. By merging isomorphism with affine transformations, for S-Box, and merging affine transformation with Inverse isomorphism in Inverse S-Box, applying composite field arithmetic of GF(((22)2)2) with factoring techniques in Multiplicative inversion, optimization will be achieved and gate count may be reduced to 300 only, and faster in speed.

First map the data to a composite field using isomorphism function δ , then compute the multiplicative inverse and re-map the computed results back to original data format using δ-1 inverse isomorphism.. The cost of isomorphism remains hidden since isomorphism is merged with affine transformation.

The α and β are roots of the primitive irreducible polynomial given below

$$P(x) = x8 + x4 + x3 + x2 + 1 \quad (4)$$

Let the generator element α in A and a generator element β in B, the isomorphism function δ (or δ-1) is immediately determined., where α k is mapped to β k (or β k to α k for any $1 \leq k \leq 254$. The hardware implementation of these functions can be obtained by mapping only the basic elements of A (or B) into B (or A). These mappings are described as multiplications of constant matrixes over GF (2).

The isomorphism functions δ and δ-1 can be determined. Computing the multiplicative inverse to reduce the cost we have to go for repeating degree -2 extensions of polynomial using irreducible polynomials.

GF (22) : $x2 + x + 1$

GF ((22) 2) : $x2 + x + \varphi$ (2)

GF (((22)2)2) : $x2 + x + \lambda$

Where φ = {10}2, λ = {1100}2. The inverter over the field has fewer GF(2) operators compared to

GF (24) : x2 +x +1

GF ((24(2) : x2 +x + ω14 (3)

Where ω14 = {1001}2 .

Multiplicative inverse cost can be further reduced when we keep n=2 and m=4 then

P-1 = (P17)-1. P16 (4)

P17 is obtained by multiplying P by P16 over GF (((22)2)2) and hardware costs for computing 2 power over Galois Field are very small ,so P16 can be calculated with less cost, and hence Inverse P.

OPTIMIZED CFA BASED COMPACT S-BOX

A design for optimizing composite field architecture for achieving high throughput for S-Boxes has been proposed (Uskov, A., Byerly, A., & Heinemann, C. 2016). There are eight possible isomorphic mappings, a common sub-expression elimination algorithm may be developed to choose mapping with minimal implementation area cost. Through algebraic normal form and fine-grained pipelined designing architecture, however a short critical depth is also desired in chips architecture, deep sub-pipelining for improved performance is also desired. S-box of AES with shortest critical path was proposed by (Standaert, F. X., Peeters, E., & Quisquater, J. J. 2005), but it required a large area compared to Canright's S-Box. (Uskov, A., Byerly, A., & Heinemann, C. 2016) proposed the optimal s-box with the shortest possible critical depth with reduced silicon area. Mapping of field, basis representation, mapping of isomorphic and field polynomials are four focus of point for CFA optimizations. All the eight possible isomorphic mappings must be examined.

Fine-grained pipelining to the GF (24) multiplier was applied by Wong to improve the performance of CFA based S-box by using AND, and XOR

operations in algebraic normal form (ANF) representation, to achieve minimum area cost by applying common sub-expression elimination algorithm to reduce area in isomorphic mapping. High throughput for FPGA implementation has been suggested (Kim, J., Hong, S., Sung, J., Lee, S., Lim, J., & Sung, S, 2003) using fully pipelined AES algorithm.

A sequence of algorithm and architectural optimization for each composite field construction is to be verified. There are eight possible isomorphic mappings, common sub-expression elimination algorithm may be developed to choose mapping with minimal implementation area cost. Through algebraic normal form and fine-grained pipelined designing architecture, we can achieve a 3.0 Gbps throughput in FPGA chip. The smallest CFA based S-box was proposed (Canright, D. 2005), however a short critical path is also desired in chips architecture, deep sub -pipelining for increased performance is also desired. S-box of AES with shortest critical path was proposed by (Standaert, F. X., Peeters, E., & Quisquater, J. J. 2005), but it required a large area compared to Canright's S-Box. The optimal s-box with the shortest possible critical path with minimum silicon area was proposed by (Uskov, A., Byerly, A., & Heinemann, C. 2016). Mapping of field, basis representation, mapping of isomorphic and field polynomials are four focus of point for CFA optimizations. All the eight possible isomorphic mappings must be examined. Wong worked on Sub sharing optimization in matrix multiplication, common sub expression algorithm to reduce area in isomorphic mappings.

Fine-grained pipelining to the GF (24) multiplier was applied by Wong to improve the performance of CFA based S-box by using AND, and XOR operations in algebraic normal form (ANF) representation, to achieve minimum area cost and highest throughput.

DERIVATION OF MULTIPLICATIVE INVERSE IN S-BOX ALGORITHM USING SUBFIELDS IN CFA

While performing S-box function during encryption of data on blocks of bytes in GF (28) with field polynomial g(x) = x8 +x4 +x3 +x +1, lot of resources are consumed in finding a multiplicative inverse and affine transformation

operation, Rijmen suggested higher order field conversion to subfields to reduce complexity, simply calculations.

First map elements of field P to a composite field Q using isomorphism function

$$q = f(p) = \delta \times p;$$

Second, compute the multiplicative inverse over Q

$$X = q\text{-}1 \text{ (except if } q = 0, \text{ then } x = 0);$$

Third, remap the computation results to P, using the inverse isomorphism function

$$P = f`(x) = \delta\text{-}1 \; X \; x$$

In mapping a computation from one field to another field to find most efficient implementation scheme, the Galois Field GF (28) is mapped to GF (((22)2)2) it requires 3 stages of isomorphism and field polynomials as follows.

a (y) = y2 + ay +v (isomorphism for GF(28) / GF(24)) (1)

b (z) = z2 + T z + N (isomorphism for GF(24) / GF (22) (2)

c (w) = w2 +w +1 (isomorphism for GF (22) / GF (2)) (3)

For equation (1) and (2) above we have to determine all the possible coefficients of v, t, n, and T in both normal and polynomial bases. Polynomial basis representation used by (Standaert, F. X., Peeters, E., & Quisquater, J. J. 2005), normal basis representation was used (Canright, D. 2005).

w2 + w +1 = 0 is the irreducible polynomial over GF (2).

OPTIMUM ISOMORPHIC MAPPING WITH COMMON SUB EXPRESSION ELIMINATION

The binary 8X8 matrix-vector product in isomorphic mapping can be expressed as eight bit- level equations. CSE would be useful in extracting common factors from these bit-level equations to reduce the area cost and the critical path.

We manually coded the circuit using a hardware description language for all of the three proposed CFA S-boxes. WE employ ANF representation along with a strategic fine grained pipeline registers insertion, in an attempt to validate the feasibility of compact CFA AES S-boxes.

Smart cards are vulnerable to differential power analysis attacks (Biryukov, A., Khovratovich, D., & Nikolic, I. 2009) and (Blömer, J., Guajardo, J., & Krummel, V. 2004), because of the limited resources, by using statically analysis of power consumption, or electromagnetic radiation to decode information to find secret key. However first order side channel attacks are overcome by adding a random mask to the data. Mask randomizes the statistics of calculation at the cost of computing the mask corrections. S-Box is a nonlinear step in each round of AES algorithm involving a Galois inversion and has high cost for mask corrections. Additive mask is maintained throughout the Galois inverse calculation by "tower field" representation (Noo-Intara, P. 2004). (Canright, D. 2005) optimized S-Box architecture proposed by Satoh by carefully selecting normal bases, which resulted in making it very compact.

Masking the data during calculation, by adding or multiplying by some random values is one countermeasure against side-channel attacks. Calculation of the mask correction is linear except Galois field inversion sub step of the S-box, in around of AES. Multiplicative mask was thought of for masking but zero data byte is unmask able by multiplication. Inversion in GF (28) involves more multiplications and one inversion in the subfield GF ((24), in turn involve further multiplications and in GF (22.). Inversion is identical to squaring, and so is linear, additive masking of Galois inverse was applied to compute mask corrections for the tower field approach (Noo-Intara, P. 2004), showed how multiplication can be eliminated by clever re-use of parts of the input mask for the output. Compact S-box of (Canright, D. 2005) applied optimization methods for mask correction terms. Some multiplication and additions were eliminated with further simplifications at lower levels, and achieved intermediate results independent of plaintext and key. It was ensures that no first order differential side-channel attacks can succeed at algorithm level. Higher-level effort will be needed for higher- order attacks.

(Lu, J., Dunkelman, O., Keller, N., & Kim, J. 2008) and (Mala, H., Dakhilalian, M., Rijmen, V., & Modarres-Hashemi, M. 2010) claimed that DPA attacks may succeed masked S-box with CMOS chips, attack exploits glitches in the gate transition timings, and he suggested that these masked S-boxes can be again made secure against first-order DPA by using more expensive logic versions.

The size of masked S-box is almost 3 times the unmasked; the speed will also be reduced. Certain applications with resources to unroll the round loop and if re-using masks between rounds can reduce some calculations. The masked S-box size can be brought down to twice the unmasked. Applications with limited resources can protected against first-order differential attacks.

CHAPTER – 5

ANALYSIS OF AES ON FIELD PROGRAMMABLE GATE ARRAYS CHIPS

CHAPTER – 5 ANALYSIS OF AES ON FIELD PROGRAMMABLE GATE ARRAYS CHIPS

FPGA SCHEMES

FPGA implementation schemes have low development cost and requires less development time. The flexibility in design variations is available if required in implementation stage; the security may be moderate to high. The developmental time is low and marketing time is short. The research proposal will deal with an FPGA implementation of AES encryption/decryption with key size of 256 bits, simulation, synthesis reports will be generated, and the results will be compared with the implementations done in the past by other researchers. Our research proposal will have key expansion module to generate round keys calculated as per the general guidelines. Our proposal is to use lookup table approach implementation for S-box to obtain high throughput by data pipeline for all rounds to achieve low latency as well.

HIGHLY SECURE AND FAST AES ALGORITHM IMPLEMENTATION IN FPGA WITH 256 BIT KEY

The Block cipher AES is a symmetric key cryptographic standard used for transferring block of data in secure manner for server based communication networks, for Gas, and Oil Pipelines, and Oil refinery and Smart Electric Grids based SCADA System applications. High security of data transfer needs long key size i.e. 256 bits, analysis of certain ideas of round key expansion mechanisms from given key data are discussed, the implementation in FPGA configuration with 128 bits and 256 bits key size to achieve low latency, high throughput with high security.

In AES encryption, the input plain text and output cipher text with a block size of 128 bits and can be viewed as a 4x4 matrix of 16 bytes arranged in a column major format. It can use a key size of 128, 192, or 256 bits and correspondingly has 10, 12 or 14 iterations of round transformations respectively. Each round transformation has four sub transformations namely; Byte Substitution (BS),

Row Shift (RS), Mix Column (MC), and Add Round Key (AK). In the last round Mix Column (MC) transformation is not included.

The key expansion mechanism is used to derive round keys from user defined cipher key as per key schedule. The total number of expanded key bytes required for a complete cipher run is equal to the no. of block length bytes (Nb) multiplied by the number of rounds (Nr) plus one. i. e. Nb(Nr+1). Thus, the total number of expanded key bytes for key size of 128, 192, and 256 bits is going to be 176, 192, and 240 bytes respectively. The increasing of a given secure key to 256-bit size results in increasing the total no. of possible codes from 2128 to 2256 and in turn good secured codes also increases accordingly. The brute force code breaking time will also be increased. The key expansion mechanism for 256 bits key size is more secure for implementation using FPGA will be discussed in this paper.

Highly secure AES algorithm implementation in FPGA data system is needed to protect data transmission between SCADA Control Server and Corporate Server of our critical integrated Corporate Industries of Petroleum, Electric Power Grids, Information Centre, Sever water control Infrastructures from cyber-attacks of national enemies, terrorist and disgruntled employees. FPGA implementation scheme for AES algorithm has been chosen because of its low system development cost and development time, in turn has short marketing time for a product, in comparison to ASIC system designs. The product can be updated for improved performance by reprogramming its software since FPGA has the flexibility in redesign variations in FPGA. An FPGA implementation is better than general-purpose processors (GPPs) as well as the application specific integrated circuits (ASICs), for developing new product application. FPGA scheme has wider applications than ASICs because its configuring software has reconfigurable nature of FPGAs. This scheme is also faster hardware solution than a GPP as proposed by (Elbayoumy, A. D., & Eldemerdash, H 2011), (Kim, J., Hong, S., Sung, J., Lee, S., Lim, J., & Sung, S, 2003) and (Gilbert, H., & Minier, M. 2000). FPGA based simulator accelerator proposed by (Kumar, B. P., Ezhumalai, P., & Gomathi, S. S. 2010), AES pipelined and unrolled implementation (Mentens, N., Batina, L., Preneel,

B., & Verbauwhede, I. 2005), FPGA based high speed and area efficient AES implementation by (Rizk, M. R. M., & Morsy, M. 2007).

The implementation of AES encryption/decryption with key size of 256 bits, simulation and synthesis report results are compared with the other implementations as listed in the table no.1. Our design uses key expansion module to generate round keys calculated as per theoretical calculations given in section 2 for key size of 256 bits, which matches exactly with that the key Expansion of 256 bits cipher given in NIST documents. Our design approach uses lookup table approach implementation for S-box to achieve low latency as well as high throughput and is low complexity architecture.

NOTATIONS AND NOTIONS FOR 256 BIT KEY

We use the data block size of 128 bits and key size of 256 bits here, use 14 rounds of iterations of round transformations.

Let for all round index i= 0,…, 14 and data byte index j=0,…, 14; key byte index l= 0,…,31;

X ij: j th text byte of i th round (in particular, X0j is the initial input plaintext byte and is fixed).

X15j: j th cipher text byte.

K il : i th expanded key byte of i-th round (in particular K0l is the user defined key : k0l : (k0, k1, k2, …, k31))

W [i] = i-th keyword of 32 bits.

Kn : nth key byte, n= {0, 1, 2, …, 239}

N k= (key size) /32 =256/32=8

N b= (block size) /32=128/32=4

Nr= No. of cipher rounds =14.

MODIFIED KEY EXPANSION OF 256 BIT KEY

The key expansion of 256-bit key size in AES is defined in the following manner.

The expanded key of N b*(Nr+1) =60 words is derived from the eight words of the user defined key.

The first eight words, W [0]… W [7] of the expanded key are filled with the user defined original cipher key bits stored in big endian format. The subsequent key words for all N k≤ i <(Nb*(Nr+1))i.e. 8≤ i<60 alternatively i = (8,…, 59) are given by:

$$W[i] = \begin{cases} W[i - N_k] \oplus Rotbyte\left(bs(W[i-1])\right) \oplus Rcon\,(i / N_k) & \forall i = 0(N_k) \\ W[i - N_k] \oplus bs(W[i-1]) & \forall i = 4(N_k) \\ W[i - N_k] \oplus W[i-1] & \forall i \neq 0,4(N_k) \end{cases}$$

First 4* N k (=32) bytes, defined as K0j: (k0, k1, k2… k31) of the expanded key are filled with the original 256 user defined bits in big endian format. For subsequent rounds, the expanded key bytes at n = {32… 239} are given by the following relations:

When n =0(mod 4* N k), or in particular at n= 32,64,96,128,160,192,224, the four consecutive key bytes at n to n+3 locations are obtained through:

K n = kn-32⊕ bs (kn-3) ⊕ Rc (n/32)

Kn+1 = K (n+1) -32⊕ bs ((kn-2))

Kn+2 = K (n+2) -32⊕ bs(kn-1)

Kn+3 = K (n+3) -32⊕ bs(kn-4)

When n= 4(mod 32), (or in particular n= 48, 80, 112, 144, 176, 208) the four consecutive key bytes in n to (n+3) locations are obtained through:

K n = k n-32⊕ bs [kn-4]

K n+1 = k (n+1) -32 ⊕ bs [kn-3]

K n+2 = k(n+2)-32 ⊕ bs [kn-2]

K n+3 = k(n+3) -32 ⊕ bs [kn-1]

The subsequent expanded key bytes for a particular round i.e. from (n+4) th byte to (n+31)th byte of k n, (or rest of n=33 to 239) are obtained through:

$$K\ n = kn-32 \oplus kn-4$$

These expanded key bytes can be represented in the form of round keys K I j with round index i and byte

Index j , through the following relations with original key bytes filled at i = 0 & j = 0, …, 31 in K 0 j .

Expanded key bytes for the subsequent rounds i.e. $0 \leq I < 8$ are obtained through the following relations:

$$Ki+10 = Ki0 \oplus bs\ (Ki29) \oplus Rc\ (i+1)$$

$$Ki+11 = Ki1 \oplus bs\ (Ki30)$$

$$Ki+12 = Ki2 \oplus bs\ (Ki31)$$

$$Ki+13 = Ki3 \oplus bs\ (Ki28)$$

$$Ki+14 = Ki4 \oplus bs\ (Ki29) \oplus Rc\ (i+1) \oplus K\ i\ o$$

$$Ki+15 = Ki5 \oplus bs\ (Ki30) \oplus K\ i\ 1$$

$$Ki+16 = Ki6 \oplus bs\ (Ki31) \oplus K\ i\ 2$$

$$Ki+17 = Ki7 \oplus bs\ (Ki28) \oplus K\ I\ 3$$

$$Ki+18 = Ki8 \oplus bs\ (Ki29) \oplus Rc\ (i+1) \oplus K\ I4 \oplus K\ I\ 0$$

$$Ki+19 = Ki9 \oplus bs\ (Ki30) \oplus K\ I\ 5 \oplus K\ i\ 1$$

$$Ki+110 = Ki10 \oplus bs(Ki31) \oplus K\,I\,6 \oplus K\,I\,2$$

$$Ki+111 = Ki11 \oplus bs(Ki28) \oplus K\,I\,7 \oplus K\,I\,3$$

$$Ki+112 = Ki12 \oplus bs(Ki29) \oplus Rc(i+1) \oplus K\,I\,8 \oplus K\,I4 \oplus K\,I\,0$$

$$Ki+113 = Ki13 \oplus bs(Ki30) \oplus K\,I\,9 \oplus K\,I\,5 \oplus K\,i\,1$$

$$Ki+114 = Ki14 \oplus bs(Ki31) \oplus K\,I\,10 \oplus K\,I\,6 \oplus K\,I\,2$$

$$Ki+115 = Ki15 \oplus bs(Ki28) \oplus K\,I\,11 \oplus K\,I\,7 \oplus K\,I\,3$$

$$Ki+116 = Ki16 \oplus bs\{(K\,I\,12 \oplus K\,I\,8 \oplus K\,I4 \oplus K\,I\,0 \oplus bs(K\,I\,29) \oplus Rc(i+1)\}$$

$$Ki+117 = Ki17 \oplus bs\{K\,I13 \oplus K\,I\,9 \oplus K\,I\,5 \oplus K\,i\,1 \oplus bs(Ki30)\}$$

$$Ki+118 = Ki18 \oplus bs\{K\,I14 \oplus K\,I\,10 \oplus K\,I\,6 \oplus K\,I\,2\ bs(Ki31)\}$$

$$Ki+119 = Ki19 \oplus bs\{K\,I15 \oplus K\,I\,11 \oplus K\,I\,7 \oplus K\,I\,3 \oplus bs(Ki28)\}$$

$$Ki+120 = Ki20 \oplus K\,i+116$$

$$Ki+121 = Ki21 \oplus K\,1+117$$

$$Ki+122 = Ki22 \oplus K\,i+1\ 18$$

$$Ki+123 = Ki23 \oplus K\,i+119$$

$$Ki+124 = Ki24 \oplus K\,i+1\ 20$$

$$Ki+125 = Ki25 \oplus K\,i+121$$

$$Ki+126 = Ki26 \oplus K\,i+122$$

$$Ki+127 = Ki27 \oplus K\,i+123$$

$$Ki+128 = Ki28 \oplus K\,i+1\ 24$$

$$Ki+129 = Ki29 \oplus K\,i+1\ 25$$

$$Ki+130 = Ki30 \oplus K\ i+126$$

$$Ki+131 = Ki31 \oplus K\ i+127$$

EXPANDED ROUND KEYS FOR 256 BIT KEY

Upon substituting the values in the expanded individual keys, it is observed that each round has a set of 32 bytes of the expanded key depending on the original 32 key bytes in the following pattern.

K0 to K31 are filled with the user defined key values. Subsequent key values are obtained using the following relation.

$$K32 = k0 \oplus bs\ (k29) \oplus Rc1$$

$$K33 = k1 \oplus bs\ (k30)$$

$$K34 = k2 \oplus bs(k31)$$

$$K35 = k3 \oplus bs(k28)$$

$$K36 = k4 \oplus k32$$

$$K37 = k5 \oplus k33$$

$$K38 = k6 \oplus k34$$

$$K39 = k7 \oplus k35$$

$$K40 = k8 \oplus k36$$

…

…

…

$$K47 = k15 \oplus k43$$

$$K48 = k16 \oplus k44$$

$$K49 = k17 \oplus k45$$

$$K50 = k18 \oplus k46$$

$$K51 = k19 \oplus k47$$

$$K52 = k20 \oplus k48$$

$$K53 = k21 \oplus k49$$

…

…

…

$$K63 = k31 \oplus k59$$

…

…

…

$$K239 = k207 \oplus k235$$

These 32 byte oriented expanded round key of 256 bit may be calculated, stored for immediate use for operations in Mobile handheld systems rather than using lookup tables, which will reduce memory requirements, for processing data in low end Spartan FPGA chips.

CHAPTER – 6

PROPOSED AES ALGORITHM WITH 256 BIT KEY IN FPGA IMPLEMENTATION

CHAPTER – 6 PROPOSED AES ALGORITHM WITH 256 BIT KEY IN FPGA IMPLEMENTATION

FPGA implementation of AES with 256 bit security key

Data transmission security level has been enhanced by using a secure key of 256 bit in place of 128-bit size and accordingly 240 bytes round expanded keys will be generated for fourteen rounds in place of 176 bytes for ten rounds respectively. Plain text data of 128 bits is encrypted in 14 rounds as shown in Figure6.1 on left side and cipher text data is decrypted using the same set of round key but using in reverse order for decryption. For data encryption operation, in round one to round, thirteen we perform BS, SR, MC, and AK transformation during each round and in round, fourteen MC transformations are not included. For data decryption operation, the reverse order of rounds is followed. We perform inverse SR, inverse BS immediately after initial AK transformation using round key 14. During remaining 13 decryption rounds, the same order of inverse transformations is used, but including inverse MC transformation in the beginning of the every round with round key number in reducing order. After last of AK transformation, we get original plain text output data.

The input secret key of 256 bits is expanded into key for fourteen rounds of 256 bits each. The 256 bits secret key expansion operation is shown in Figure 6.2. The first half of 128 bits of given 256 bits security key are termed as round key0 and the second half as round key1. Round key0 is used for first AK operation with plain text data during start of encryption. Round key1 is used for AK operation during round1 of encryption. Round key2 to round key14 are generated for AK operations, for rounds 2 to 14 as shown in the figure 9. Round keys generated during encryption are stored and utilized for AK operations of decryption also but are used in reverse direction.

When start pulse is given to the controller module, clock pulse, reset pulse, enable pulse and en/de pulse are generated by controller module. Controller module sends first reset and clock pulses to key generation module and

encryption / decryption module, then send 0/1 signal to encryption/ decryption module for encryption or decryption operation depending signal level is 0 or 1 respectively.

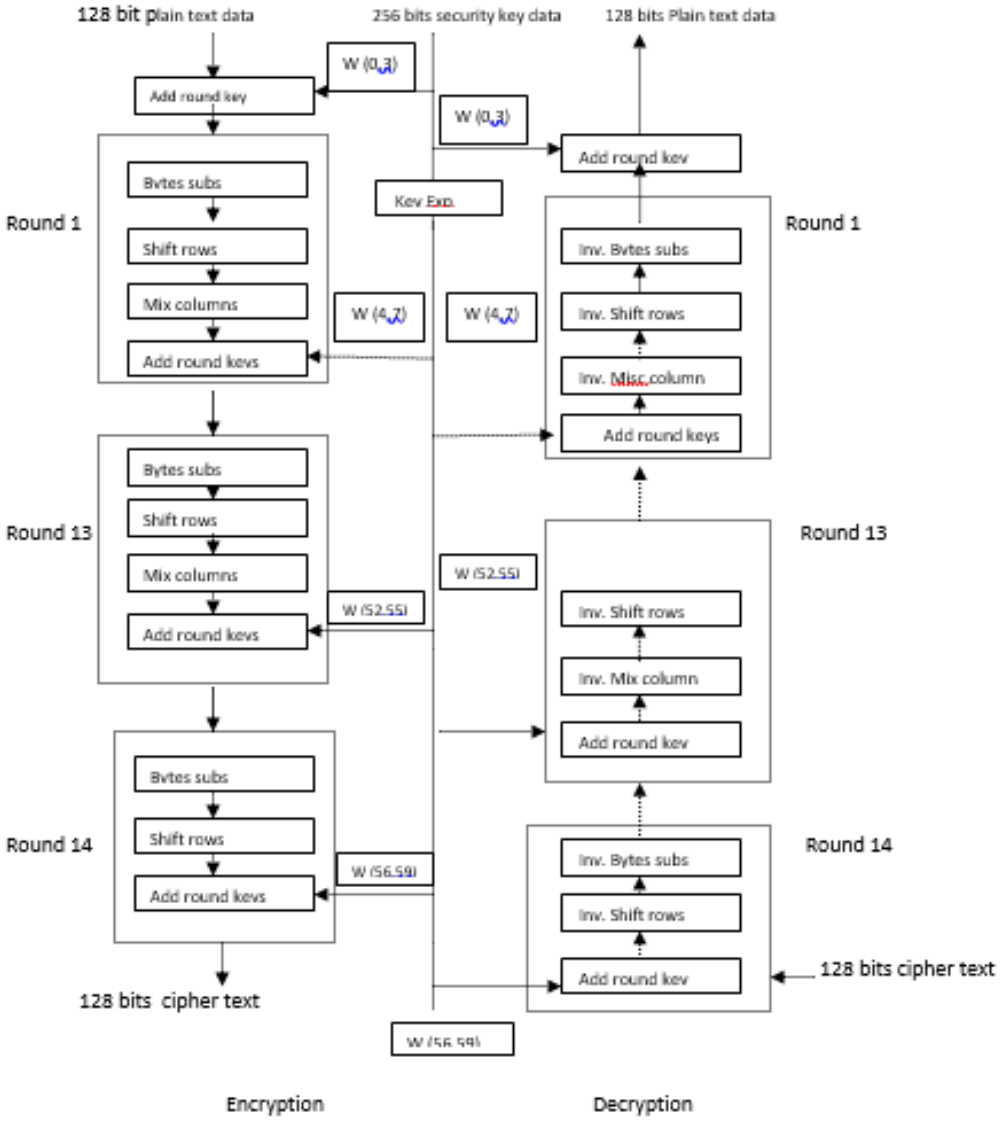

Figure 6.1 Data Encryption and Decryption with 256 bits security key

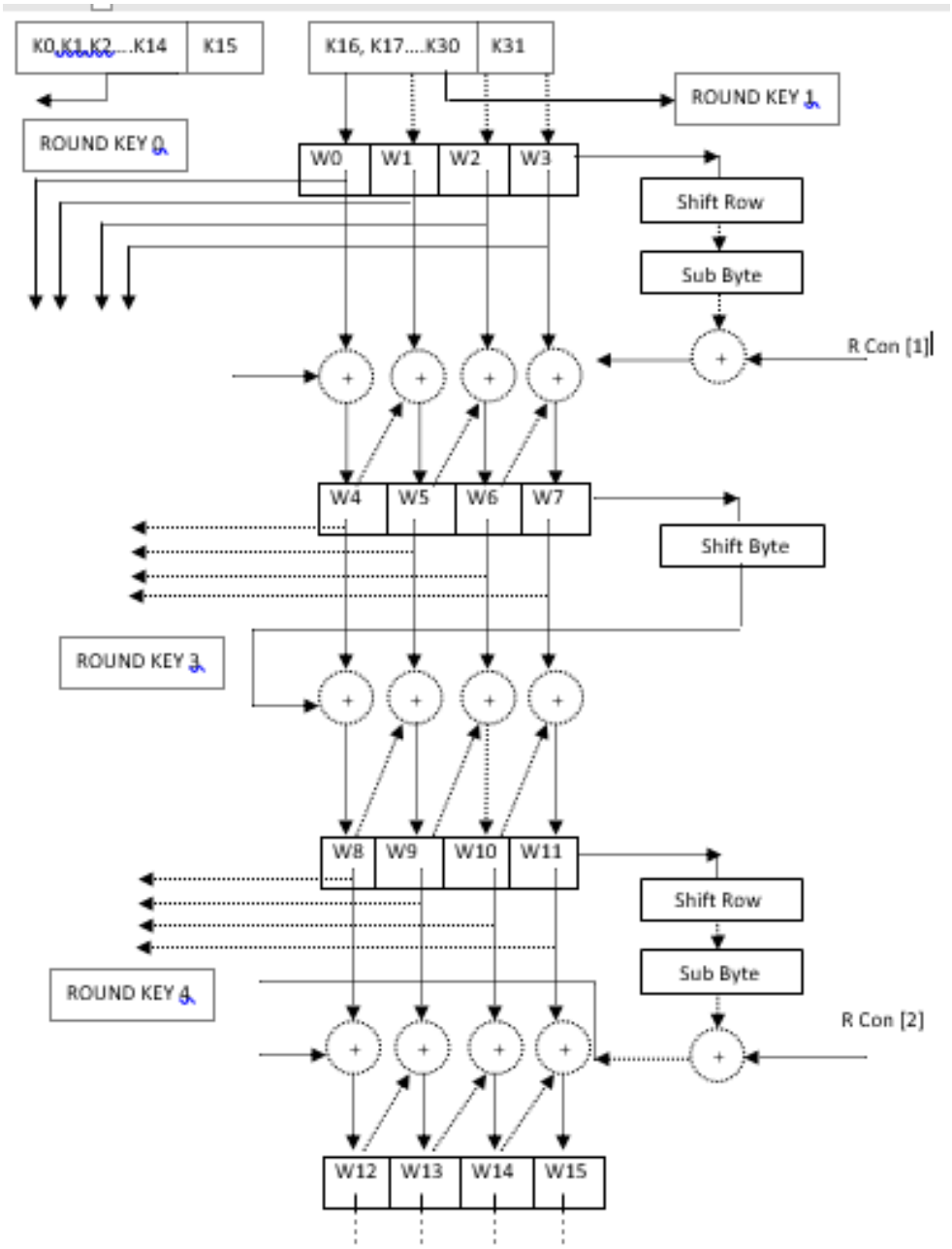

Figure 6.2 (a) 256 Bits AES Security Key Expansion Operation

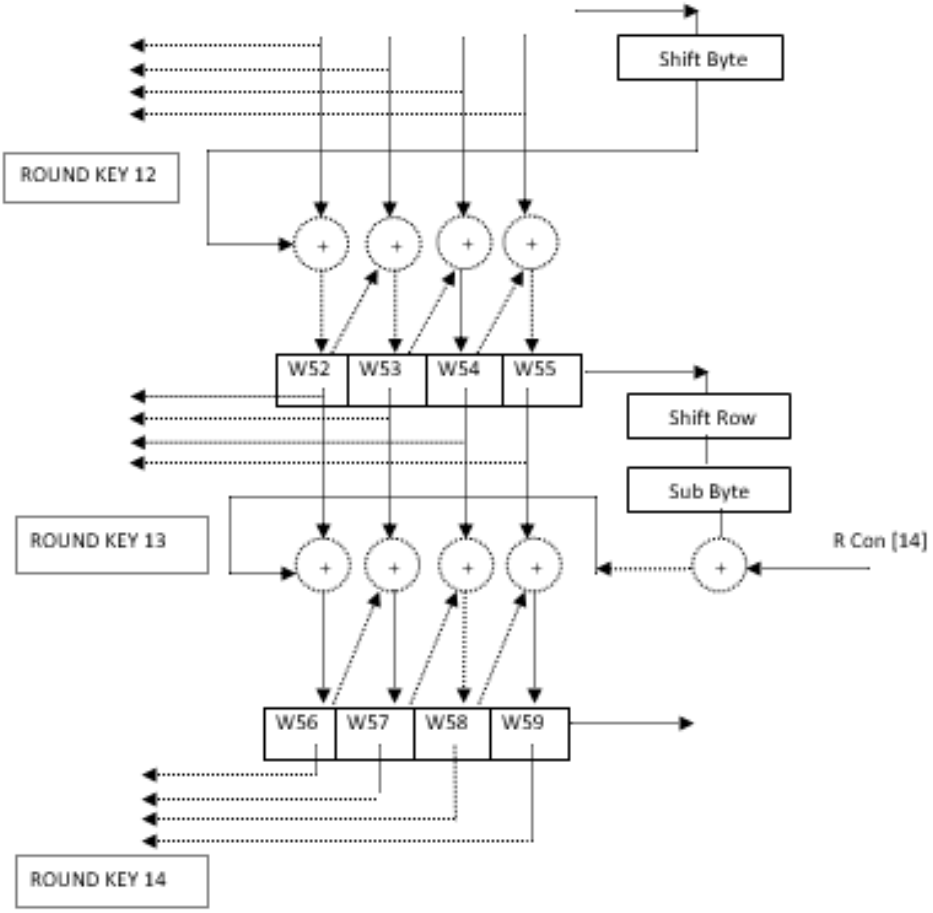

Figure 6.2 (b) 256 Bits AES Security Key Expansion Operation

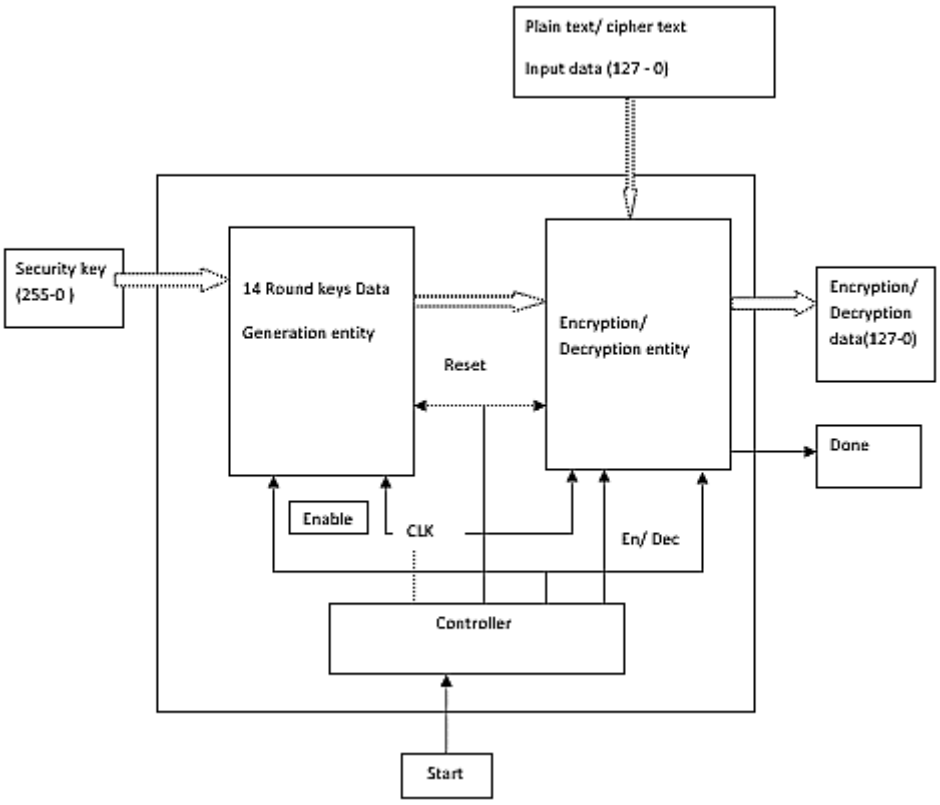

Figure 6.3 Encryption and Decryption of Top Level Entity

The input security key of 256 bits data and input plain text / cipher text of 128 bits data are entered in key generation module and encryption / decryption module, respectively, on getting enable pulse from controller module as shown in Figure 6.3. The encrypted/decrypted data of 128 bits is outputted at output port, and done pulse is generated by encryption/decryption module.

Simulation and synthesis results

The design has been coded using VHDL, all the results are synthesized based on Xilinx ISE Software 12.4 version, and target device used was xc5vtx240t-2-ff 1759. The results of simulation of encryption/decryption with security key of 256 bits with 128 bits input data, all “zeroes” value and all 128 bits of “one” value are shown in Figure 6.4 and Figure 6.5 respectively. Simulation results shows that input plain text data is properly ciphered in encryption operation and when ciphered text is given as input to decryption operation, deciphered data is found to be the original input data of encryption operation. All the round keys

generated during encryption operation are found to be the same as given in NIST documents for security key of 256 bits.

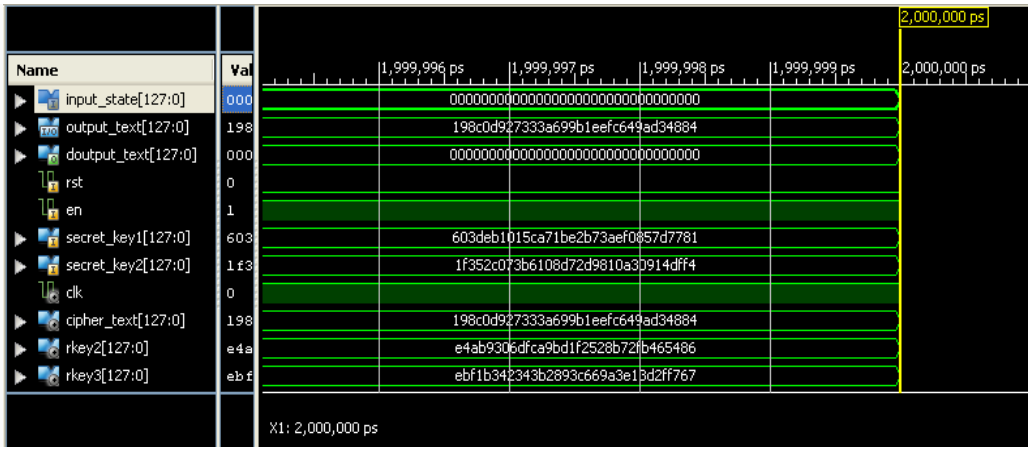

Figure 6.4 Simulation results with all the 128 input data bits as ‘’zeros’’ for 256 bits key.

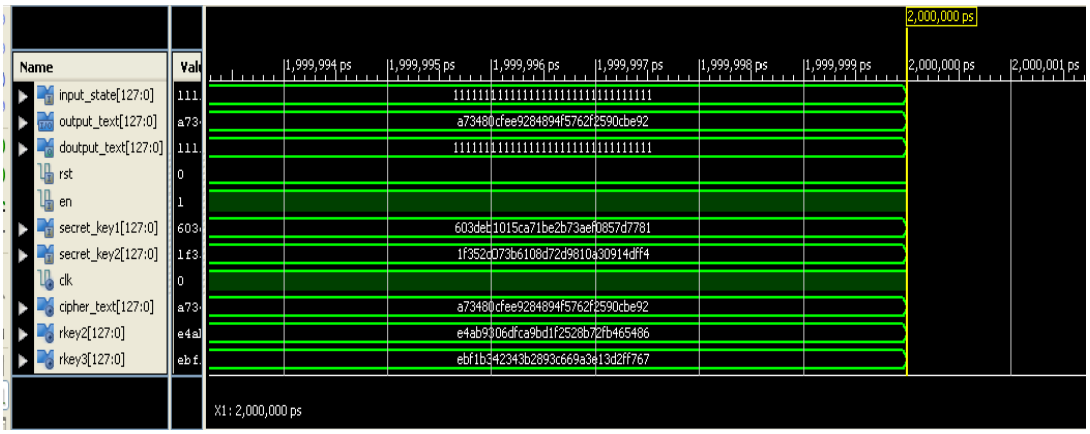

Figure 6.5 Simulation results with all the 128 input data bits as ‘’ones’’ for 256 bits key.

Synthesis report for 256-bit security key is generated for AES algorithm based on Xilinx ISE software 12.4 versions, for target device xc5vtx240-2-ff1759, the report data is given below:

1. No. of ROMs: 500
2. No. of Flip Flops: 14336
3 No. of input and output pins: 642
2. No. of Slice LUT’s: 27517

3. Clock period: 2.115nS
4. Maximum Frequency: 472.82 MHz
5. Delay: 2.115nS
6. Throughput: 64 GBPS

COMPARISONS OF RESULTS OF AES ALGORITHM WITH 128 BIT AND 256 BIT SECURITY KEYS

Two schemes of FPGA implementations of 128-bit data block size with 128 bits security key and 256 bits security key respectively have been implemented and results have been compared with results reported by other authors as shown in Table 2. The comparative table clearly shows that our pipelined architecture using lookup tables for S-blocks are better in terms of latency, throughput and higher security with 256 bits security key.

Table 6.1 Comparison of results for AES with security key of 256 bits key
Simulation of AES with 256 Bits Security Key, Input data as all 0000H:

Design	Device used	Area/Slices used	Throughput Megabits/sec	Throughput Megabits/Slice	Maximum frequency in MHz
1. K. Gaj& P. Chodowiec [33]	XCV1000BG560-6 XC2S30-6	2902 222; GRAM-3	331.5 166	----- 0.132	----- 60
2. Dandalis [ ]	XCV-1000	5673	353.0	0.062	-----
3. Elbirt et.al [46]	XCV1000-4	10992; BRAM-0	-----	-----	31.8
4. Mcloone	XCV812E-8	2000; BRAM-224	-----	-----	93.3
5. Helion	Virtex 4-11	1016	-----	-----	200.0
6. G. Rouvroy [95]	XC3S50-4	163 BRAM-3	208	1.26	71
7. Swinder Kaur [64]	Virtex2 p-7	6279; BRAM-5			119.95
8. Amandeep kaur [63]	XC2VP30-5-FF896	1127	-----	-----	247.3
9. Thulasimani [97]	XC-2V600BF-957-6	2943	666.7	0.226	-----
10. Our Design AES- 128 bits security key	XC5VTX240T-2FF1759-2	10240; BRAM-0	**4720**	**0.460**	**472.8**
11. Our Design AES- 256 bits security key	XC5VTX240T-2FF1759-2	14336; BRAM-0	**4720**	**0.329**	**472.8**

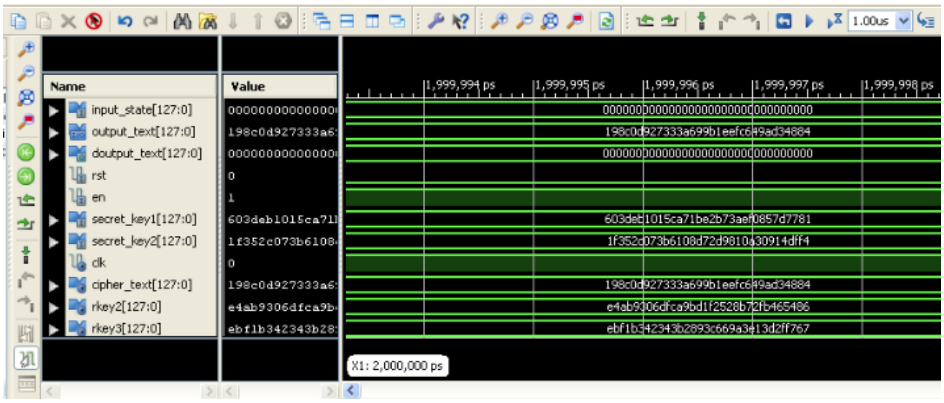

Figure 6.6 Simulation of AES with 256 Bits Security Key, Input data as all 0000H

Input data have been given all 128 bits as zeros and encrypted data produced a random data of 128 bits at transmitter output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [127-bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

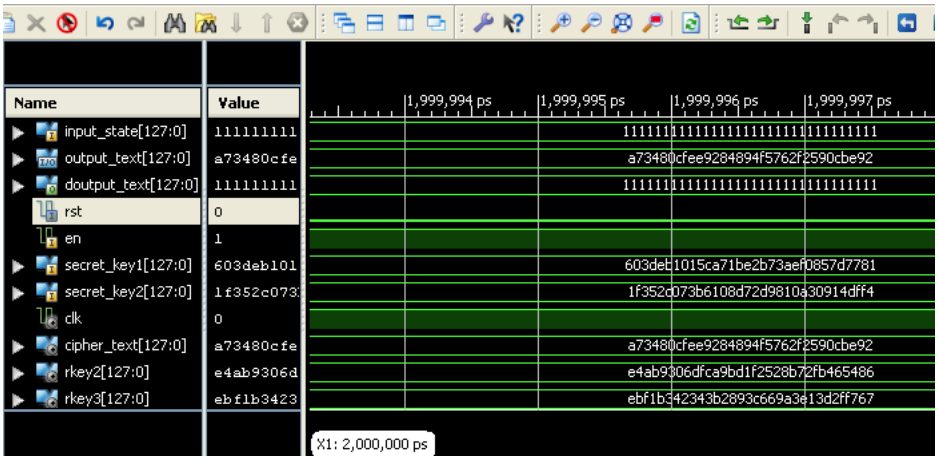

Figure 6.7 Simulation of AES with 256 Bits Security Key, Input data as all 1111H

Input data have been given all 128 bits as Ones Hex data 01H Hex data [00010001] bits and encrypted data produced a random data of 128 bits at transmitter output _text [128bits]. At receiver when the encrypted data have

been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

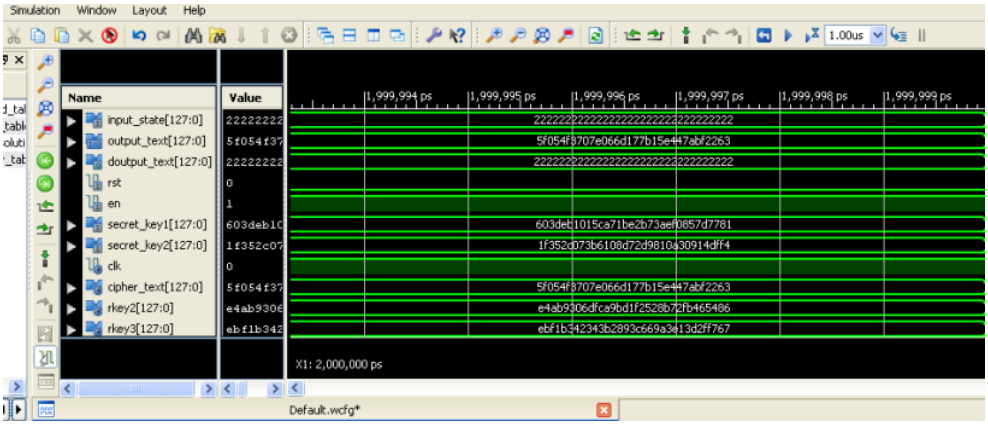

Figure 6.8 Simulation of AES with 256 Bits Security Key, Input data as all 2222H

Input data have been given all 128 bits as Twos Hex data [0010 0010 Bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is regenerated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

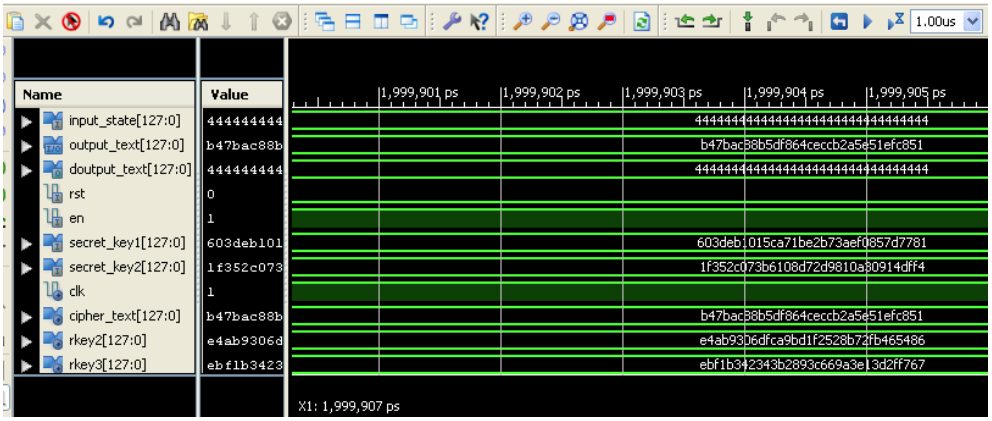

Figure 6.9 Simulation of AES with 256 Bits Security Key, Input data as all 4444H

Input data have been given all 128 bits as Fours Hex data [0100 0100 Bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

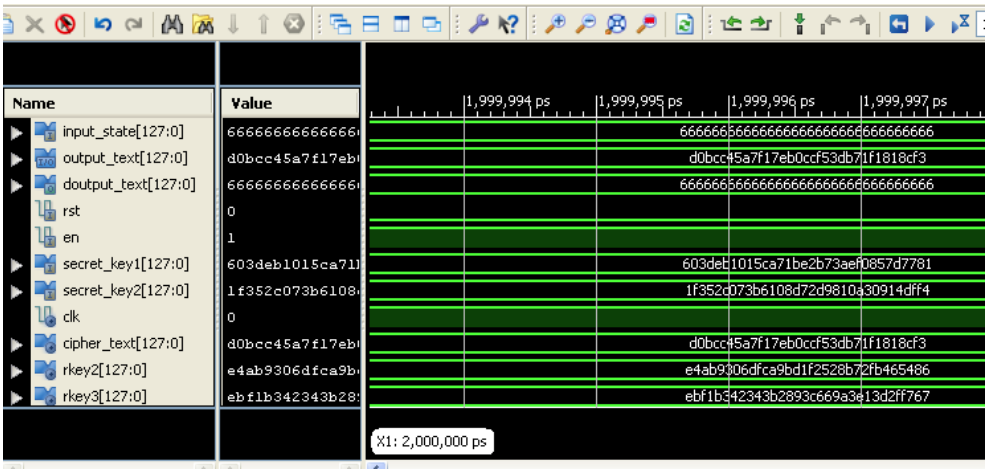

Figure 6.10 Simulation of AES with 256 Bits Security Key, Input data as all 6666H

Input data have been given all 128 bits as Sixes Hex data [0110 0110 Bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

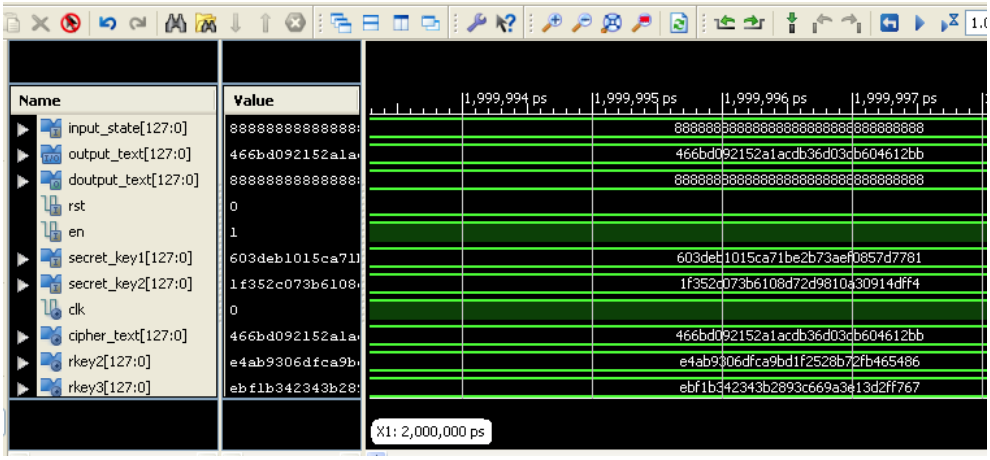

Figure 6.11 Simulation of AES with 256 Bits Security Key, Input data as all 8888H

Input data have been given all 128 bits as Eights Hex data [1000 1000 Bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

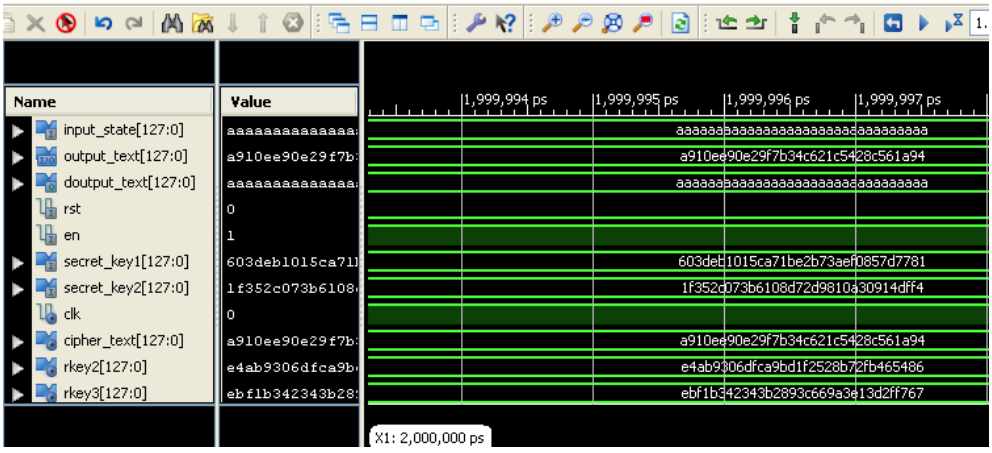

Figure 6.12 Simulation of AES with 256 Bits Security Key, Input data as all aaaa H

Input data have been given all 128 bits as aaaa Hex data [1010 1010 Bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have

been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

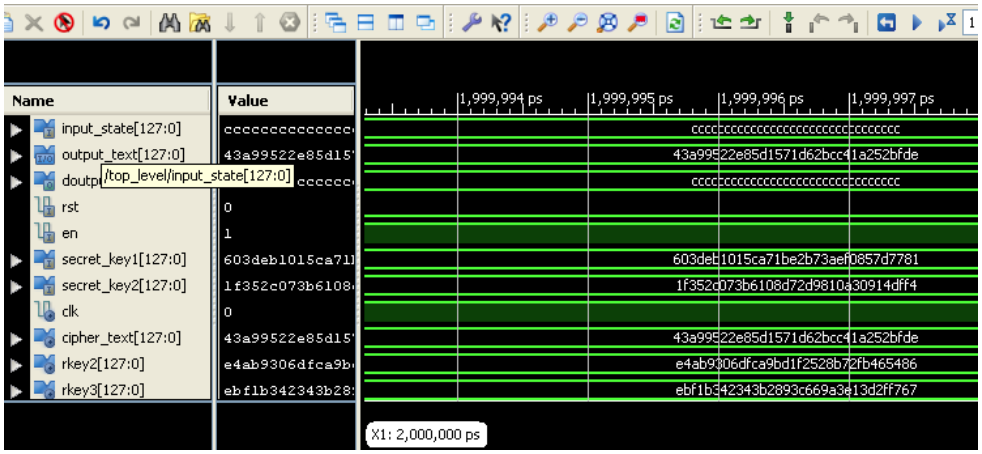

Figure 6.13 Simulation of AES with 256 Bits Security Key, Input data as all cccc H

Input data have been given all 128 bits as aaaa Hex data [1100 1100 Bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

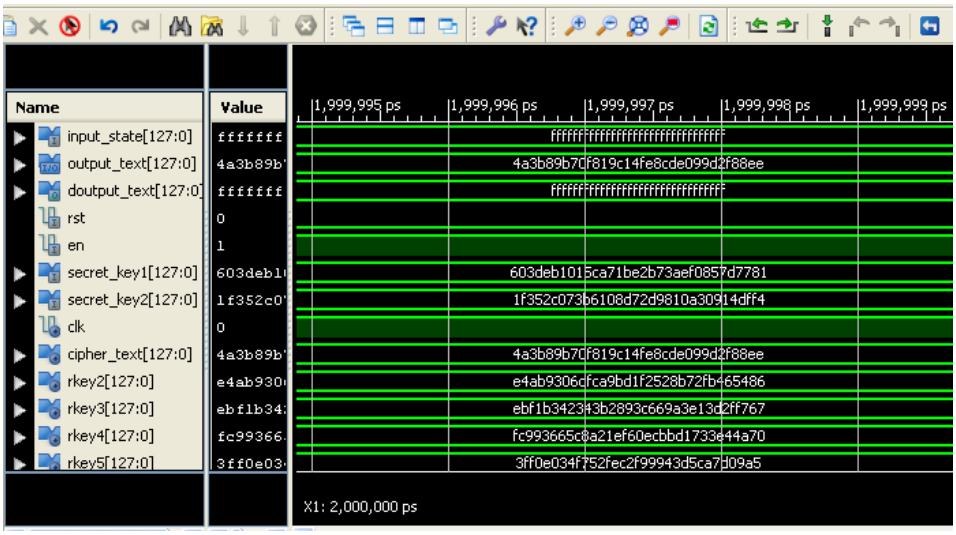

Figure 6.14 Simulation of AES with 256 Bits Security Key, Input data as all ffff H

Input data have been given all 128 bits as ffff Hex data [1111 1111 Bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is regenerated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

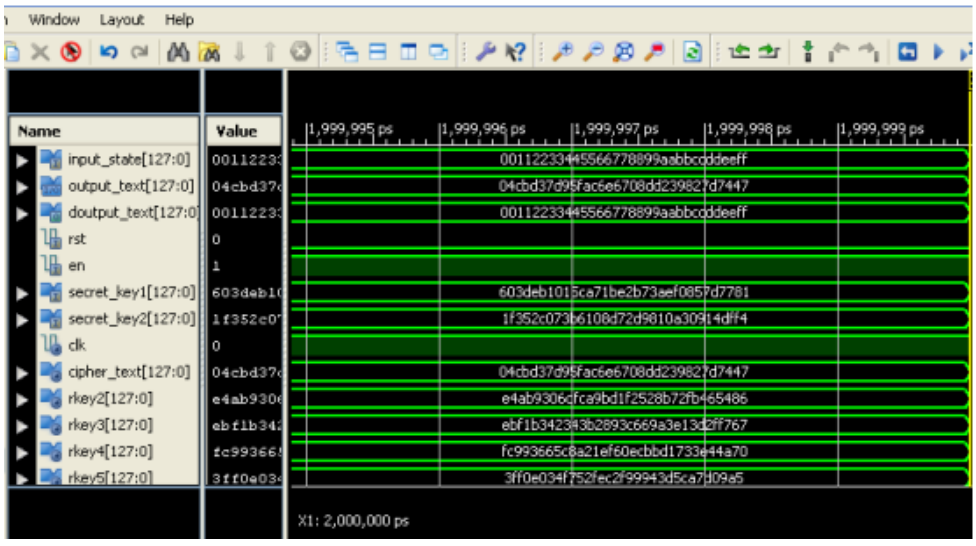

Figure 6.15 Simulation of AES with 256 Bits Security Key, Input data [00112233445566778899aabbccddeeff] H:

Input data have been given all 128 bits as 00112233445566778899aabbccddeeff Hex data [00000000 bits], [00010001 bits], [00100010 bits], [00110011 bits], [01000100 bits], [01010101 bits], [01100110 bits], [01110111 bits], [10001000 bits], [10011001 bits], [10101010 bits], [10111011 bits], [11001100 bits], [11011101 bits], [11101110 bits], [11111111 bits], and encrypted data is produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is regenerated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

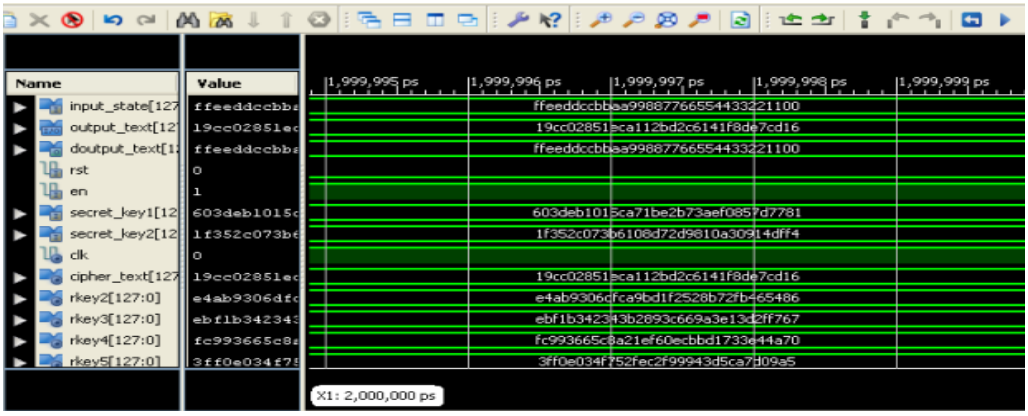

Figure 6.16 Simulation of AES with 256 Bits Security Key input data [ffeeddccbbaa99887766554433221100] h:

Input data have been given all 128 bits as [ffeeddccbbaa 99887766554433221100] Hex data [11111111 bits], [11101110 bits], [11011101 bits], [11001100 bits], [10111011 bits], [10101010 bits], [10011001 bits], [10001000 bits], [01110111 bits], [01100110 bits], [01010101 bits], [01000100 bits], [00110011 bits], [00100010 bits], [00010001 bits], [00000000 bits]. and encrypted data is produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys

for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

Simulation of AES with 256 Bits Security Key, Input data a single 1 bit at start point in Hex:

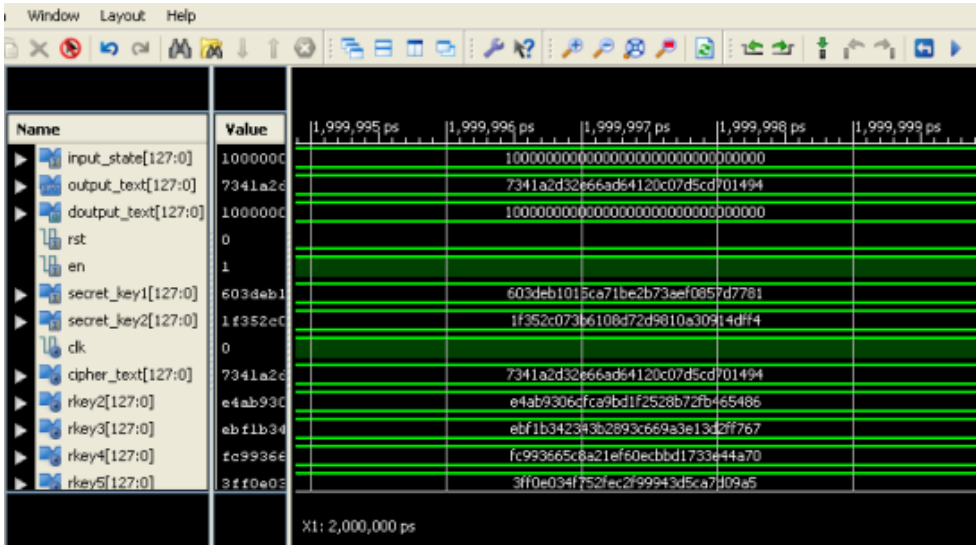

Figure 6.17 Simulation of AES with 256 Bits Security Key, Input data a single 1 bit at start Point in Hex:

Input data have been given all 128 bits as [1000 0000] Hex, [0000 0000] Hex, [0000 0000] Hex, [0000 0000] Hex data and equivalent binary data as [00010000] bits, [00000000] bits, [0000 0000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00010000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, and encrypted data is produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver. It is observed that only one bit out of 128 bits is made input as one but all intermediate data is total random at every stage.

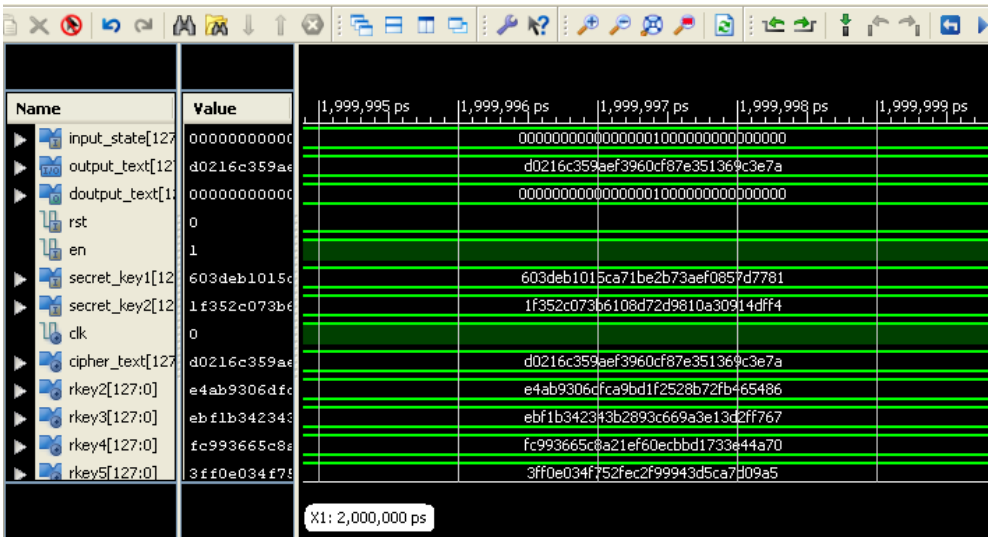

Figure 6.18 Simulation of AES with 256 Bits Security Key input data a single bit at mid-point in Hex:

Input data have been given all 128 bits as [0000 0000] Hex, [0000 0000] Hex, [1000 0000] Hex, [0000 0000] Hex data and equivalent binary data as [00000000] bits, [00000000] bits, [0000 0000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00010000] bits, [00010000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, and encrypted data is produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver. It is observed that only one bit out of 128 bits is made input as one but all intermediate data is total random at every stage.

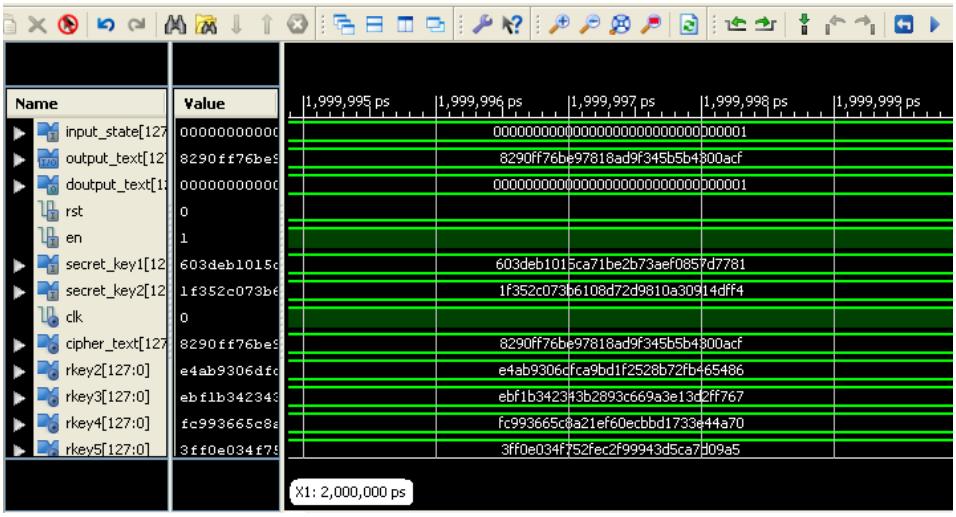

Figure 6.19 Simulation of AES with 256 Bits Security Key input data a single bit at End point in Hex:

Input data have been given all 128 bits as [0000 0000] Hex, [0000 0000] Hex, [0000 0000] Hex, [0000 0001] Hex data and equivalent binary data as [00000000] bits, [00000000] bits, [0000 0000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000001] bits, and encrypted data is produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver. It is observed that only one bit out of 128 bits is made input as one but all intermediate data is total random at every stage.

SIMULATION RESULTS CONDUCTED ON AES WITH SECURITY KEY OF 128 BITS

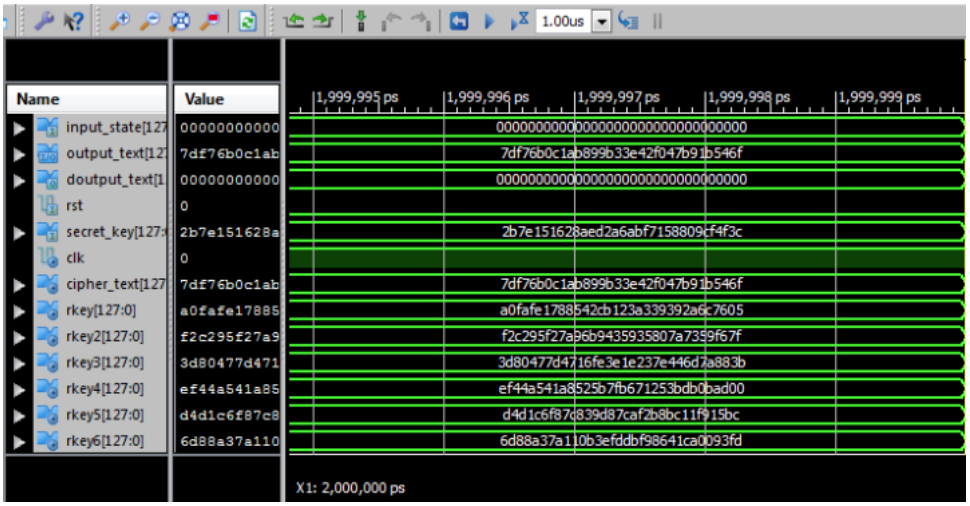

Figure 5.20 Simulation of AES with 128 Bits Security Key, Input data 0000H

Input data have been given all 128 bits as 0000 Hex data [0000 0000 Bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

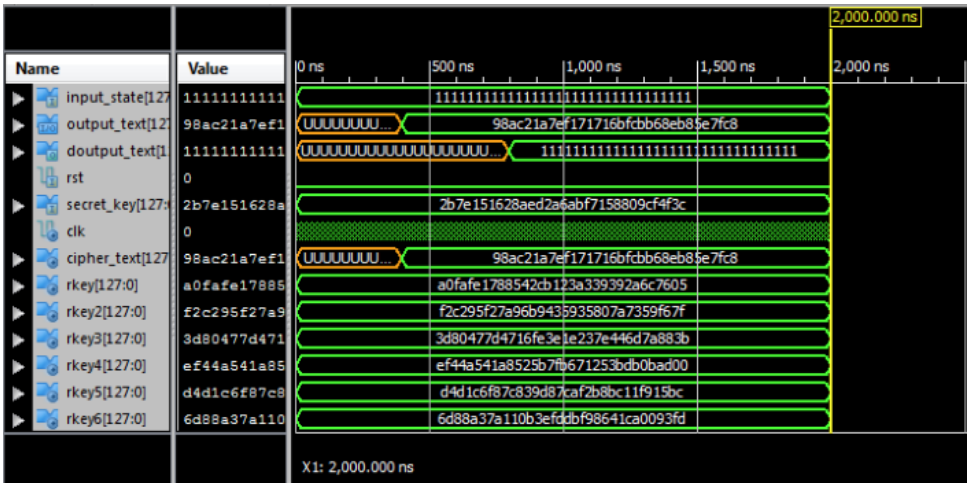

Figure 6.21 Simulation of AES with 128 Bits Security Key, Input data 1111H:

Input data have been given all 128 bits as 1111 Hex data [00010001 bits] for every byte data and encrypted data produced a random data of 128 bits at

transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

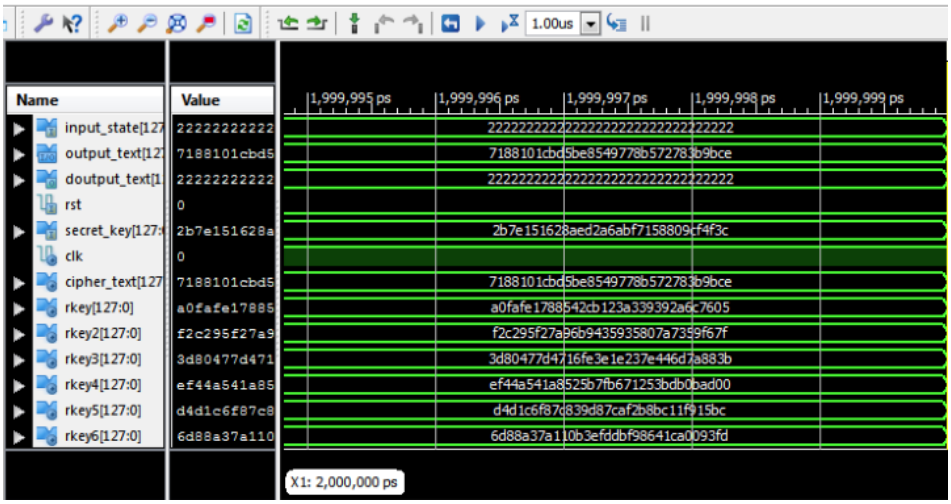

Figure 6.22 Simulation of AES with 128 Bits Security Key, Input data 2222H

Input data have been given all 128 bits as 2222 Hex data [00100010 bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

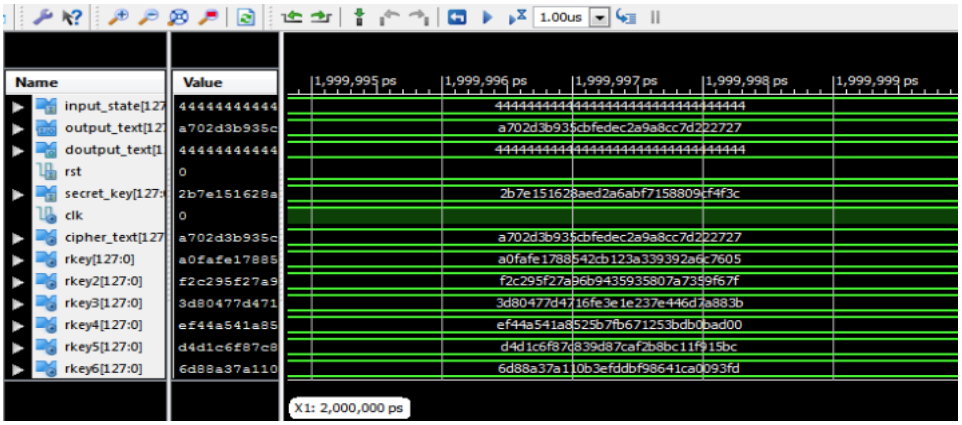

Figure 6.23 Simulation of AES with 128 Bits Security Key, Input data 4444H

Input data have been given all 128 bits as 4444 Hex data [01000100 bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

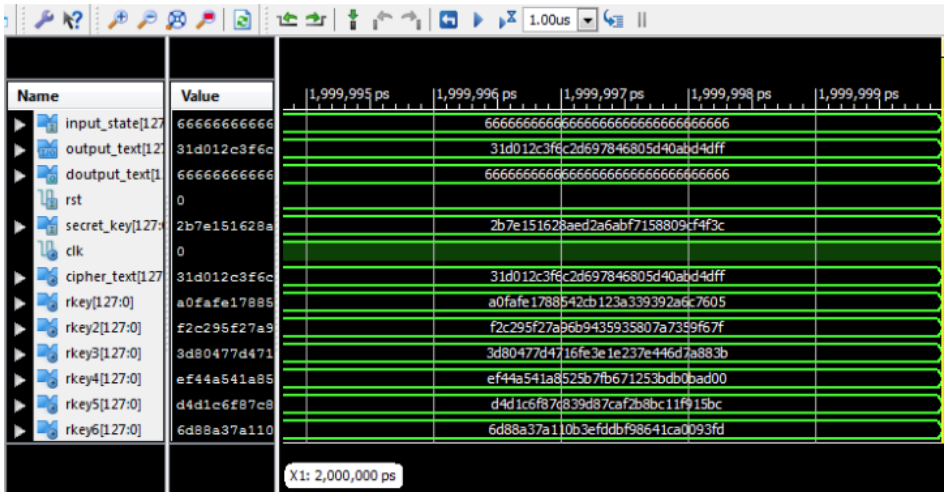

Figure 6.24 Simulation of AES with 128 Bits Security Key, Input data 6666H

Input data have been given all 128 bits as 6666 Hex data [01100110 bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

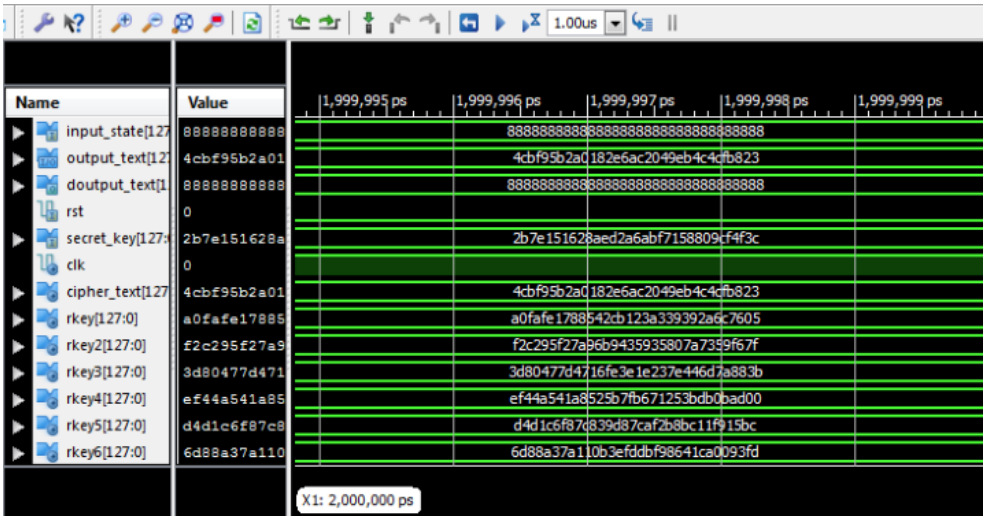

Figure 6.25 Simulation of AES with 128 Bits Security Key, Input data 8888H

Input data have been given all 128 bits as 8888 Hex data [10001000 bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

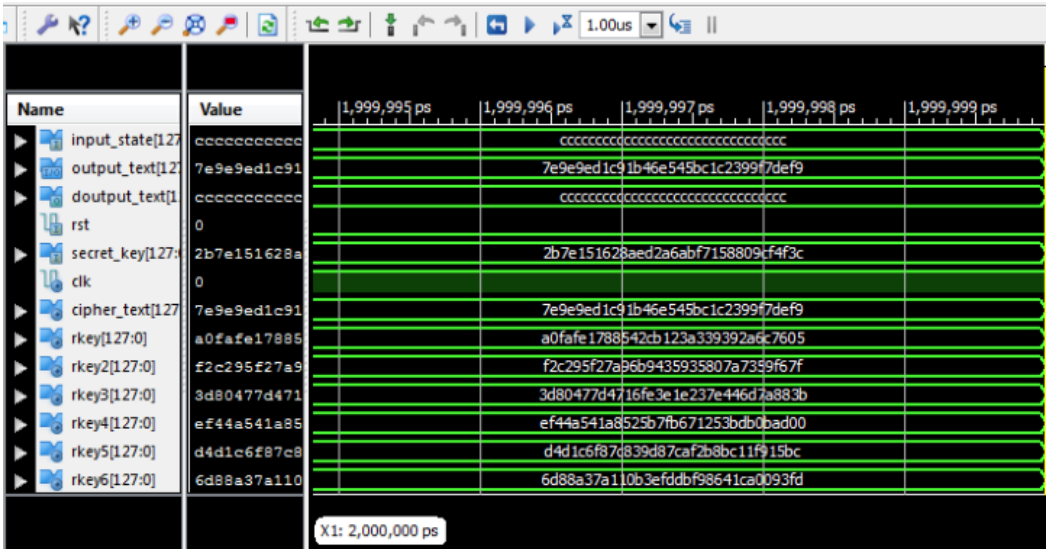

Figure 6.26 Simulation of AES with 128 Bits Security Key, Input data as cccc H

Input data have been given all 128 bits as cccc Hex data [11001100 bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

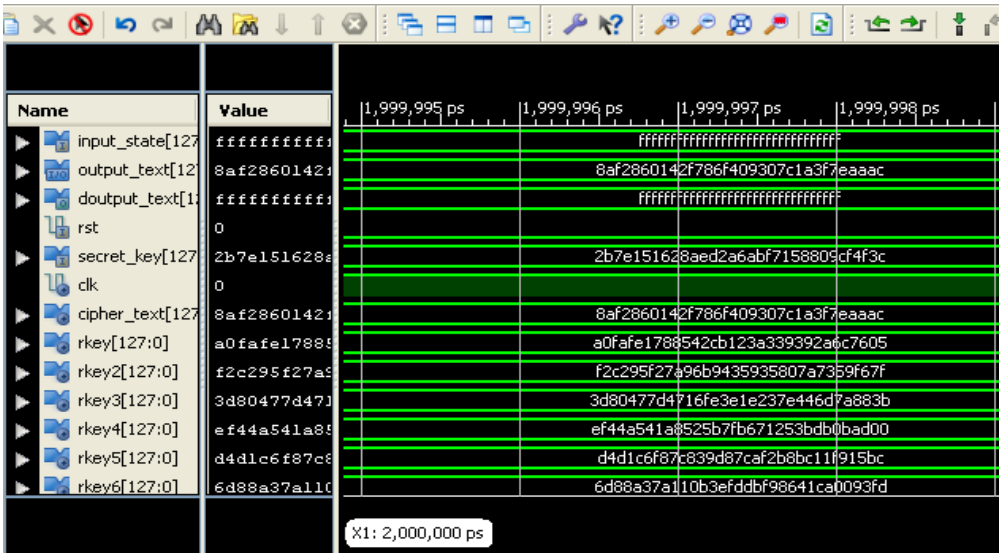

Figure 6.27 Simulation of AES with 128 Bits Security Key, Input data ffff H

Input data have been given all 128 bits as ffff Hex data [11111111 bits] for every byte data and encrypted data produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver

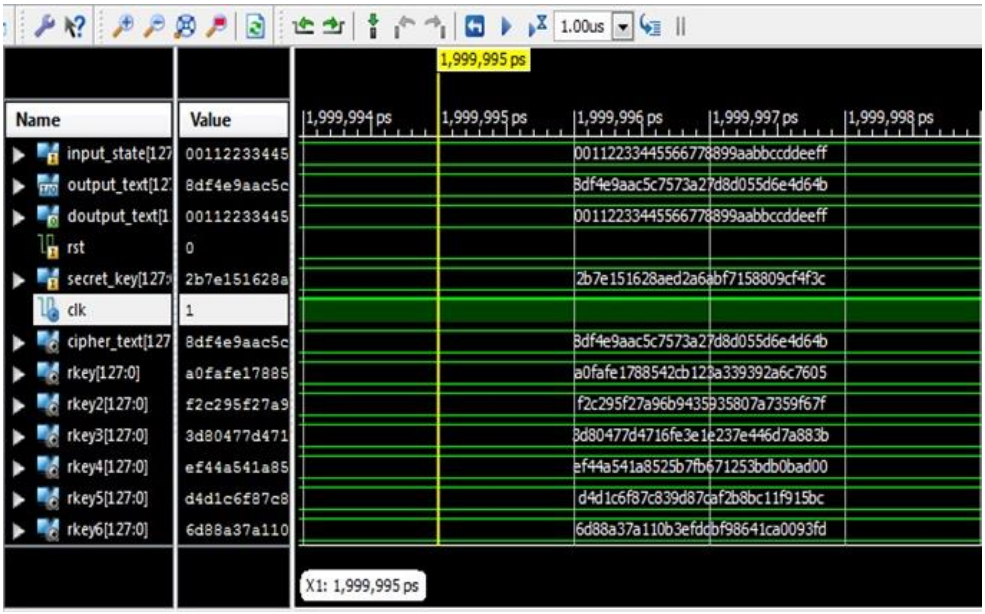

Figure 6.28 Simulation of AES with 128 Bits Security Key, Input data [00112233445566778899aabbccddeeff] H

Input data have been given all 128 bits as [00112233445566778899aabbccddeeff] Hex data [00000000 bits], [00010001 bits], [00100010 bits], [00110011 bits], [01000100 bits], [01010101 bits], [01100110 bits], [01110111 bits], [10001000 bits], [10011001 bits], [10101010 bits], [10111011 bits], [11001100 bits], [11011101 bits], [11101110 bits], [11111111 bits], and encrypted data is produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

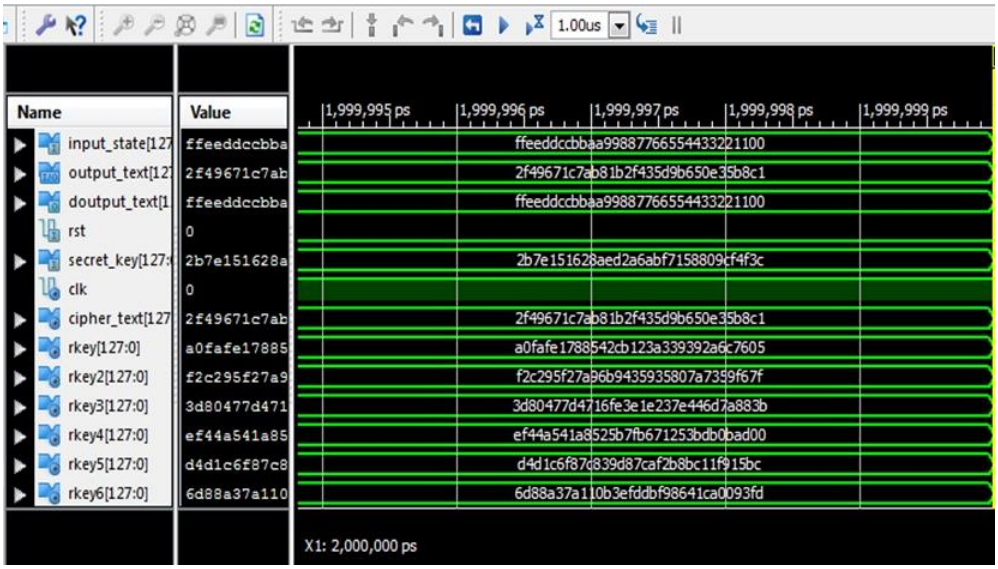

Figure 6.29 Simulation of AES with 128 Bits Security Key, input data [ffeeddccbbaa99887766554433221100]

Input data have been given all 128 bits as [ffeeddccbbaa99887766554433221100] Hex data [11111111 bits], [11101110 bits], [11011101 bits], [11001100 bits], [10111011 bits], [10101010 bits], [10011001 bits], [10001000 bits], [01110111 bits], [01100110 bits], [01010101bits], [01000100 bits], [00110011 bits], [00100010 bits], [00010001 bits], [00000000 bits], and encrypted data is produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver.

Simulation of AES with 128 Bits Security Key input data a single 1 bit at start point

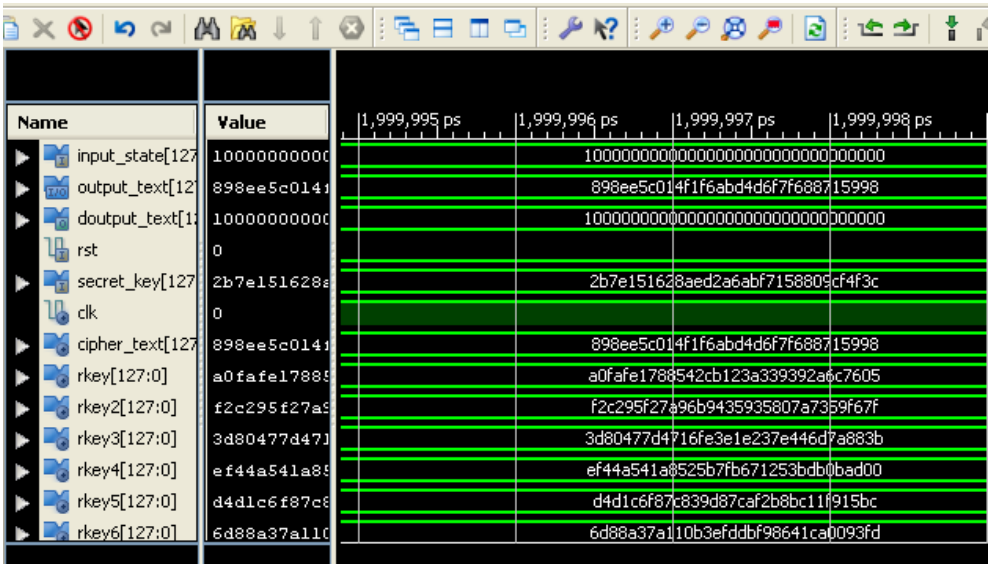

Figure 6.30 Simulation of AES with 128 Bits Security Key input data a single 1 bit at start point

Input data have been given all 128 bits as [1000 0000] Hex, [0000 0000] Hex, [0000 0000] Hex, [0000 0000] Hex data and equivalent binary data as [00010000] bits, [00000000] bits, [0000 0000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00010000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, [00000000] bits, and encrypted data is produced a random data of 128 bits at transmitter as output _text [128bits]. At receiver when the encrypted data have been given as input data, the encrypted data is deciphered and the original data of transmitter input is re generated at receiver output doutput_text [128bits]. All round keys for processing intermediate round key are generated and stored for encrypting the input data at transmitter and deciphering receiver data at receiver. It is observed that only one bit out of 128 bits is made input as one but all intermediate data is total random at every stage.

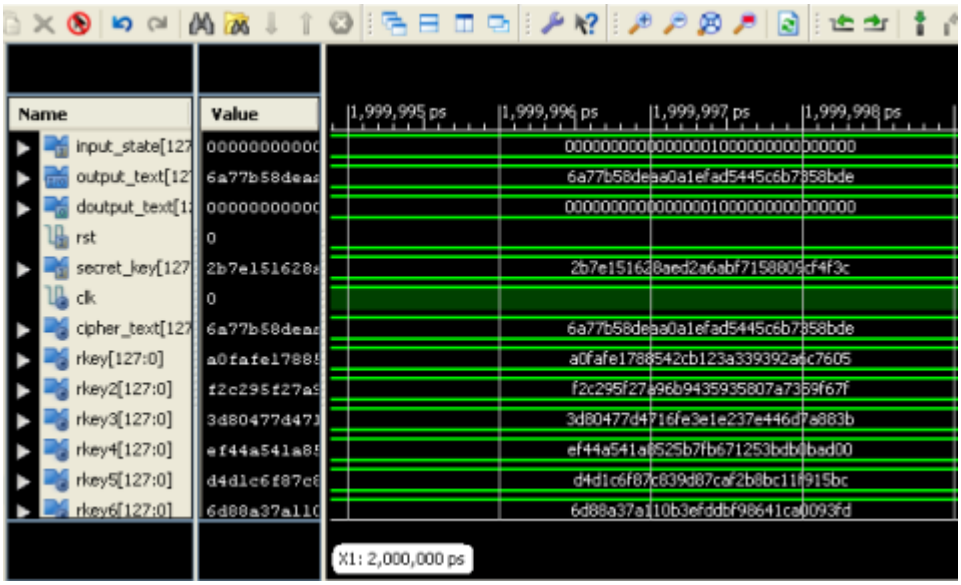

Figure 6.31 Simulation of AES with 128 Bits Security Key input data a Single 1bit at mid-point

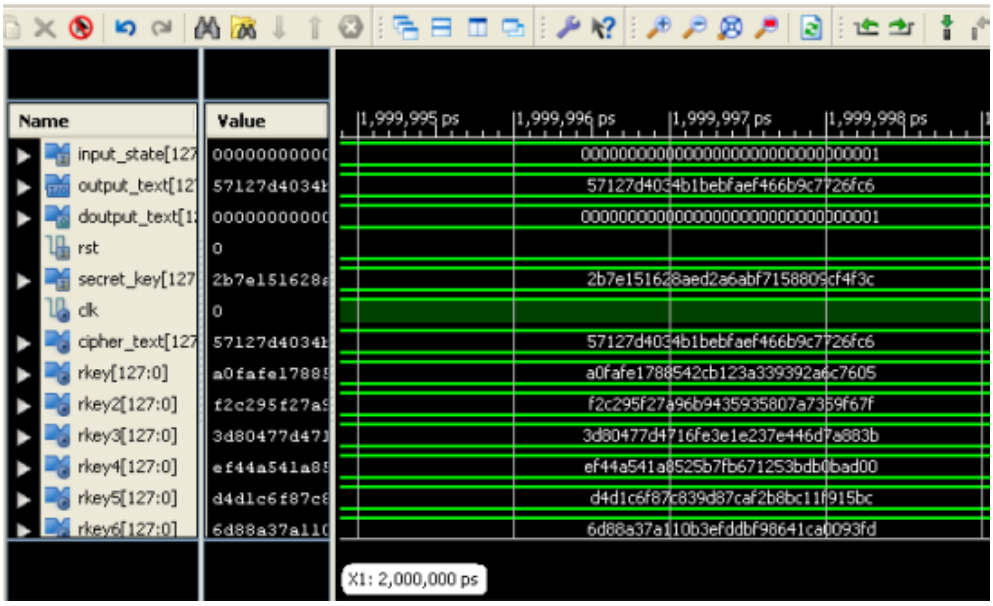

Figure 6.32 Simulation of AES with 128 Bits Security Key input data a Single bit at End Point

SIMULATION OF AES WITH 256 BITS SECURITY KEY

Input data a single 1 bit at start point, displaying all intermediate processed data of round keys and intermediate transformations data

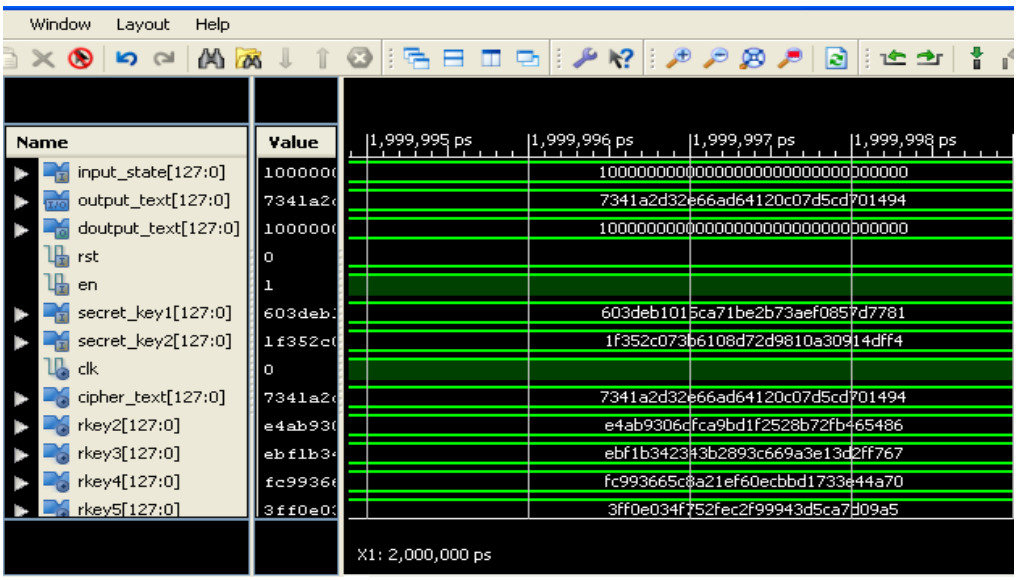

Figure 6.33 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No.A

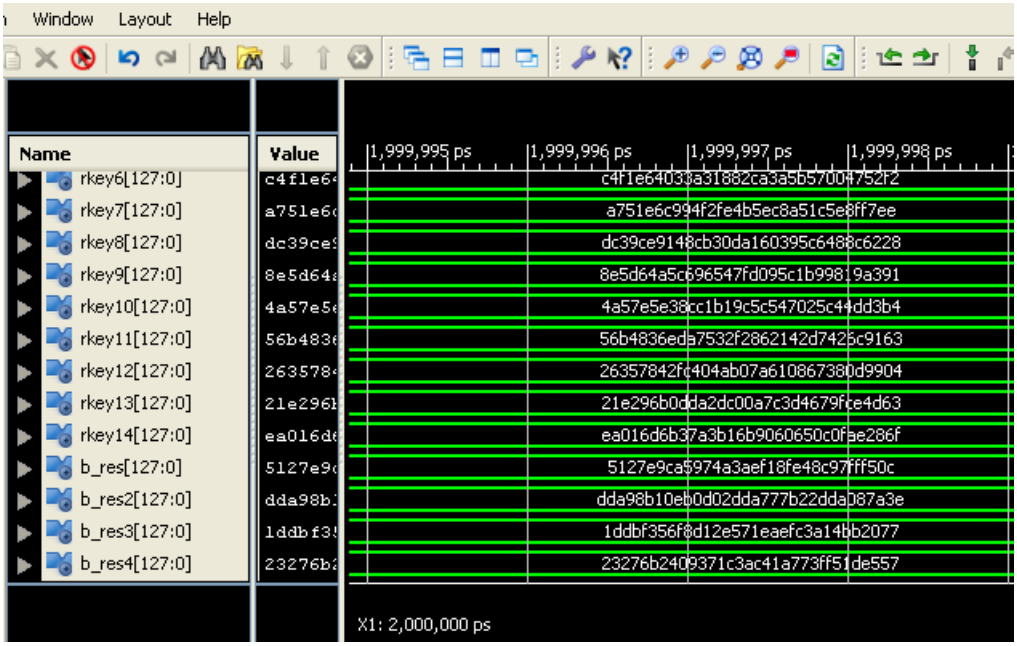

Figure 6.34 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. B

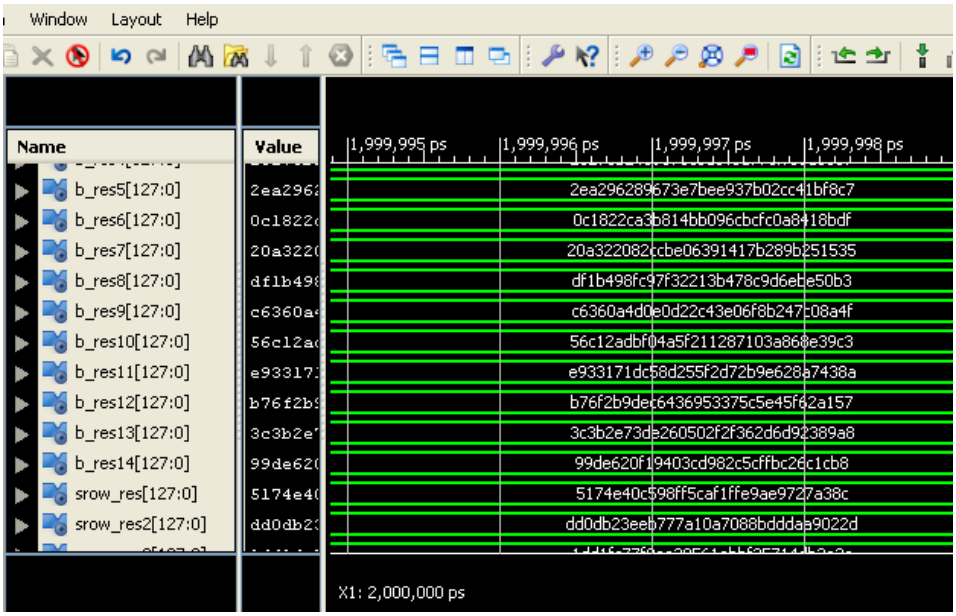

Figure 6.35 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. C

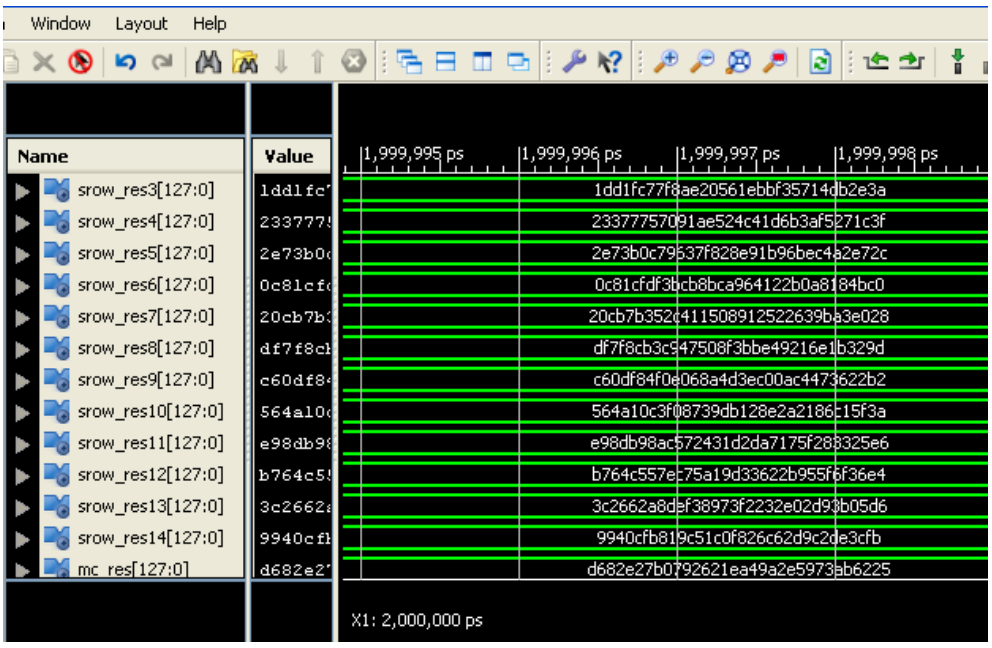

Figure 6.36 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. D

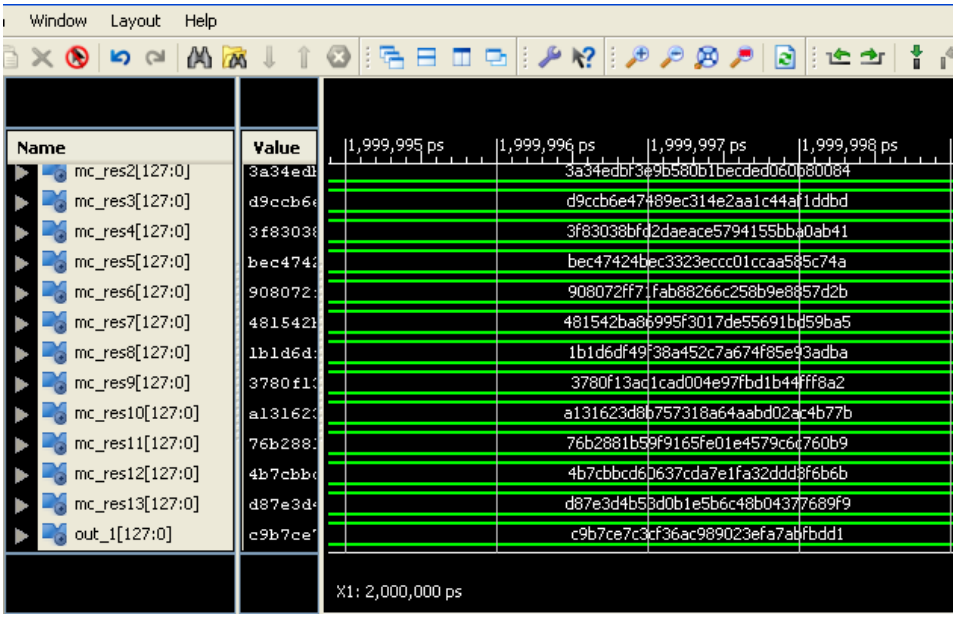

Figure 6.37 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No.E

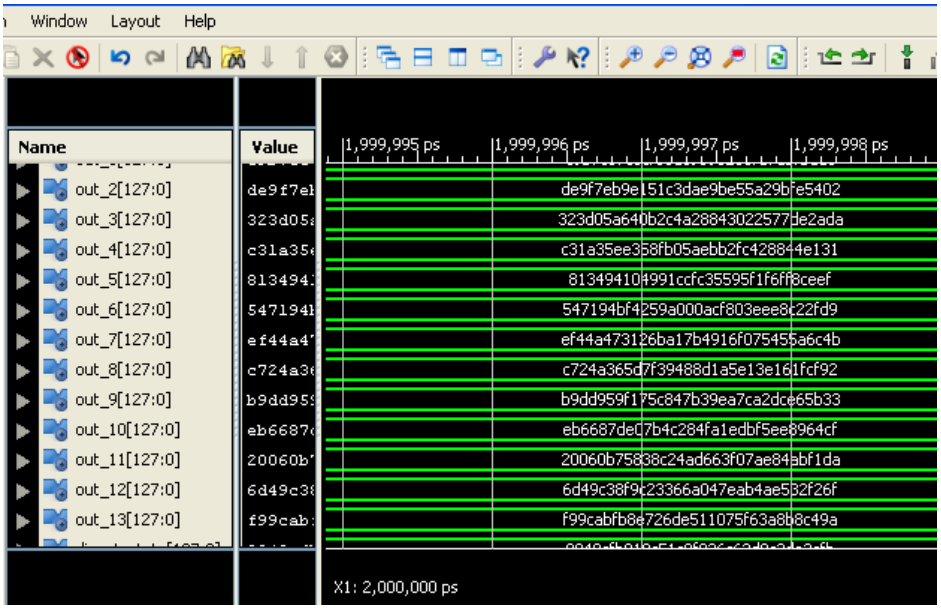

Figure 6.38 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. F

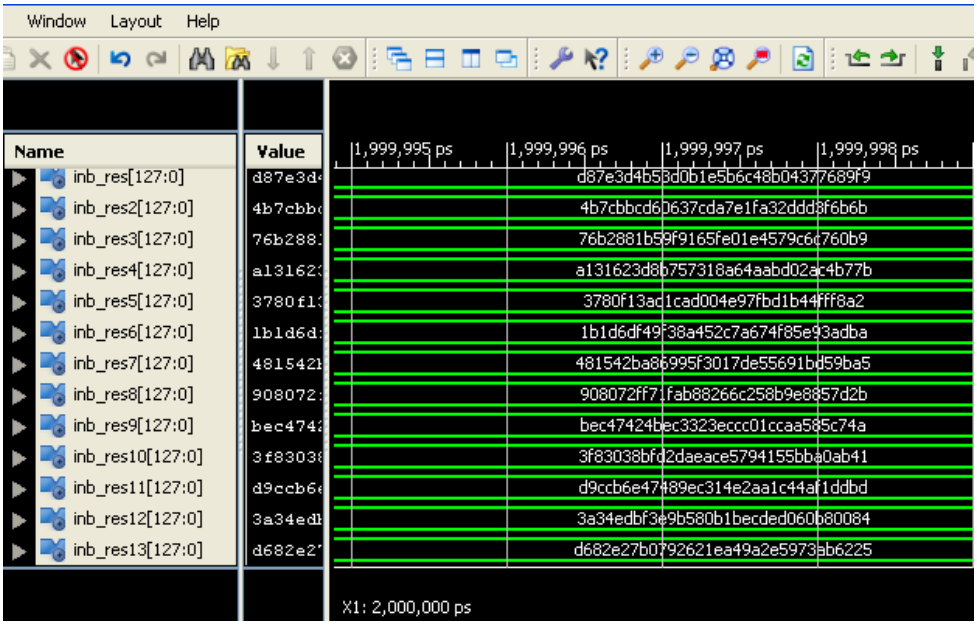

Figure 6.39 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No.G

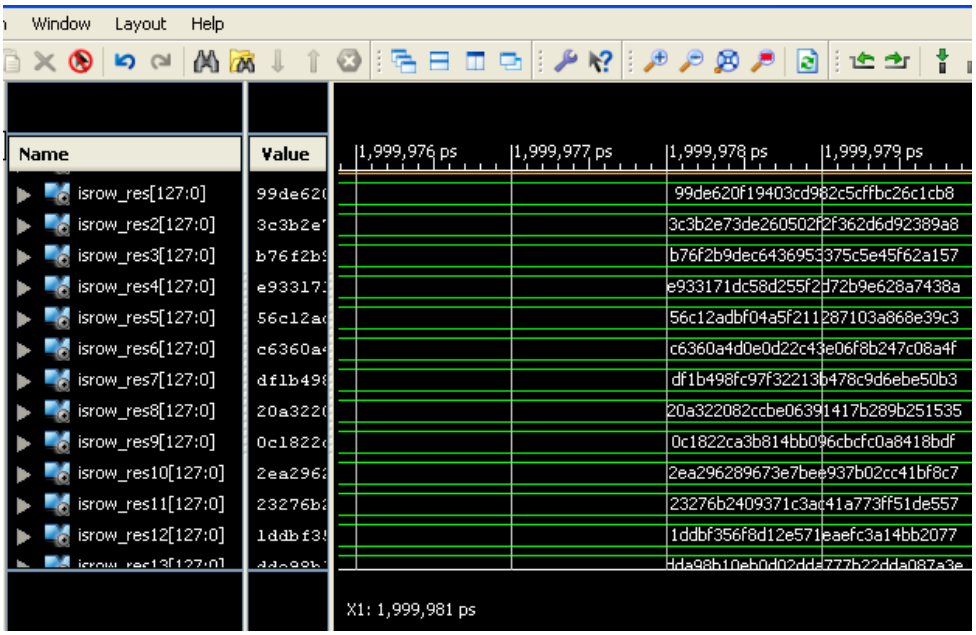

Figure 6.40 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. H

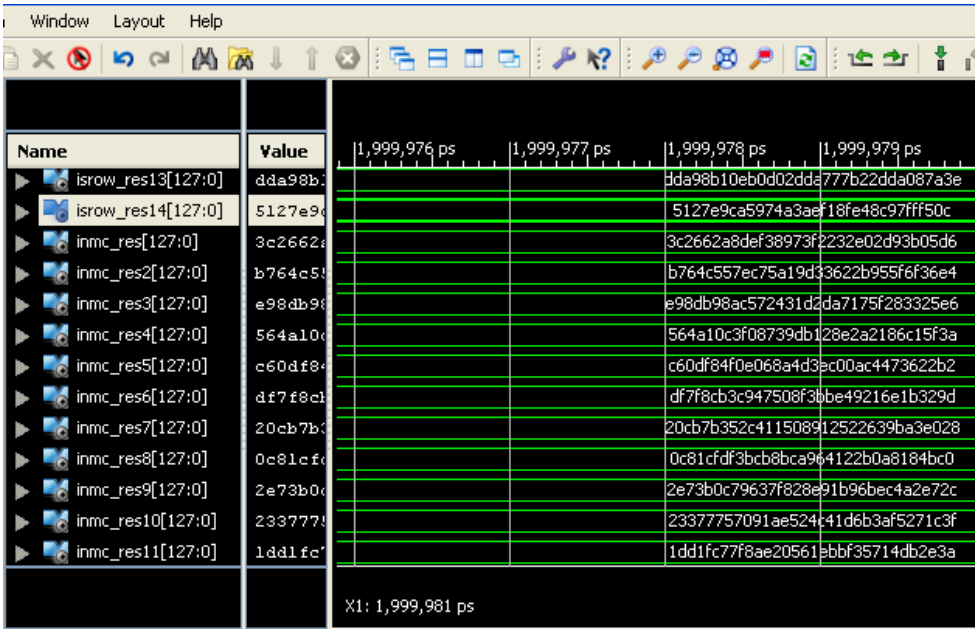

Figure 6.41 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. I

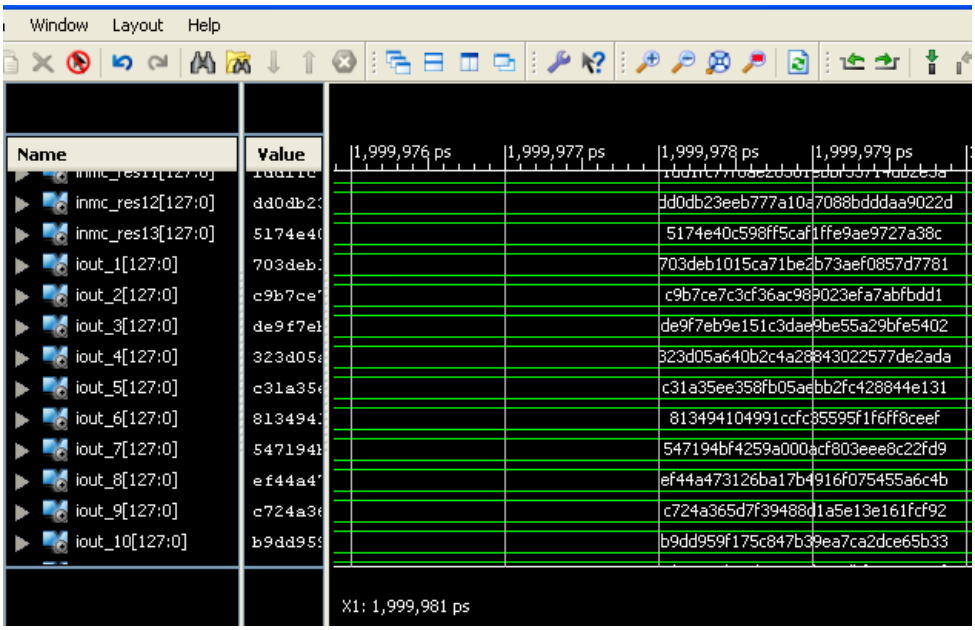

Figure 6.42 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. J

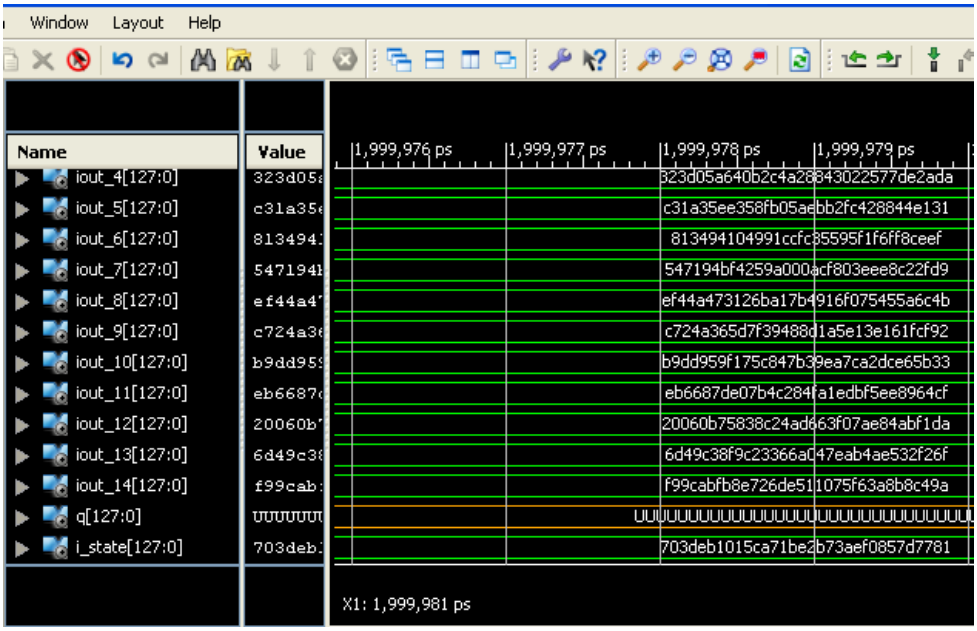

Figure 6.43 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. K

Simulation of AES with 256 Bits Security Key, input data a single 1 bit at Middle point, displaying all intermediate processed data of round keys and intermediate transformations data

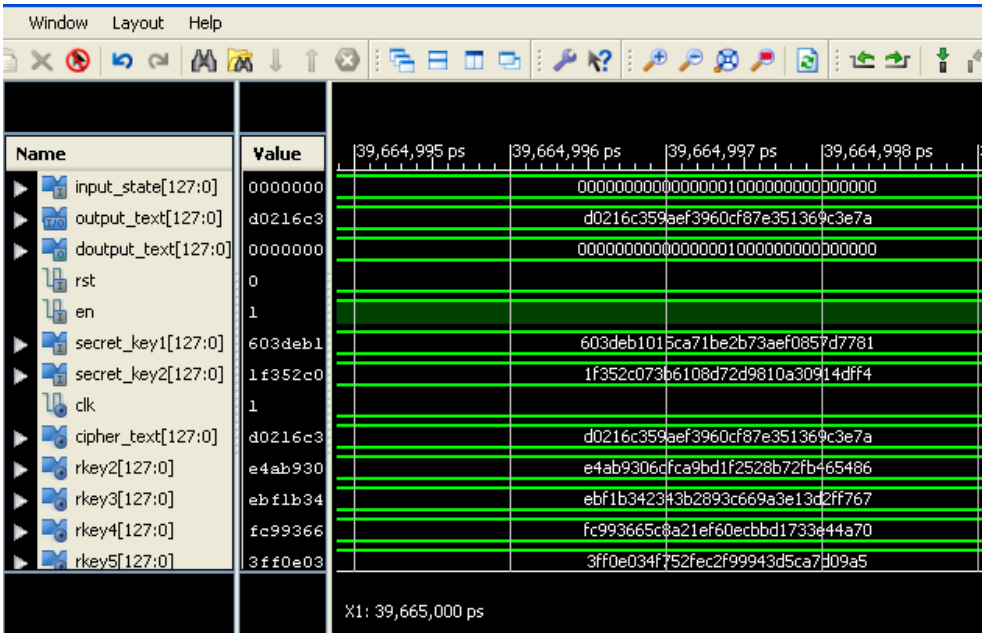

Figure 6.44 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. a

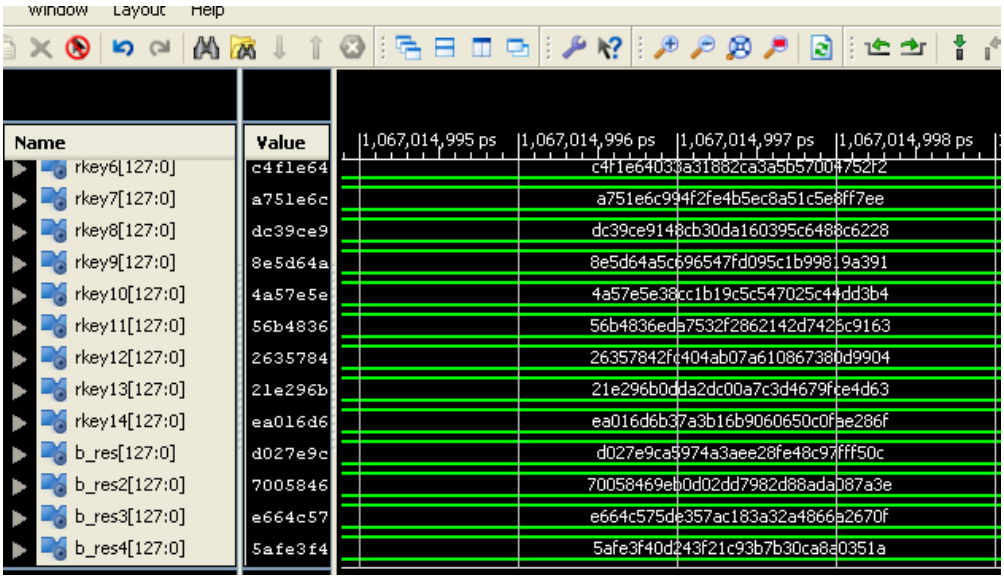

Figure 6.45 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. b

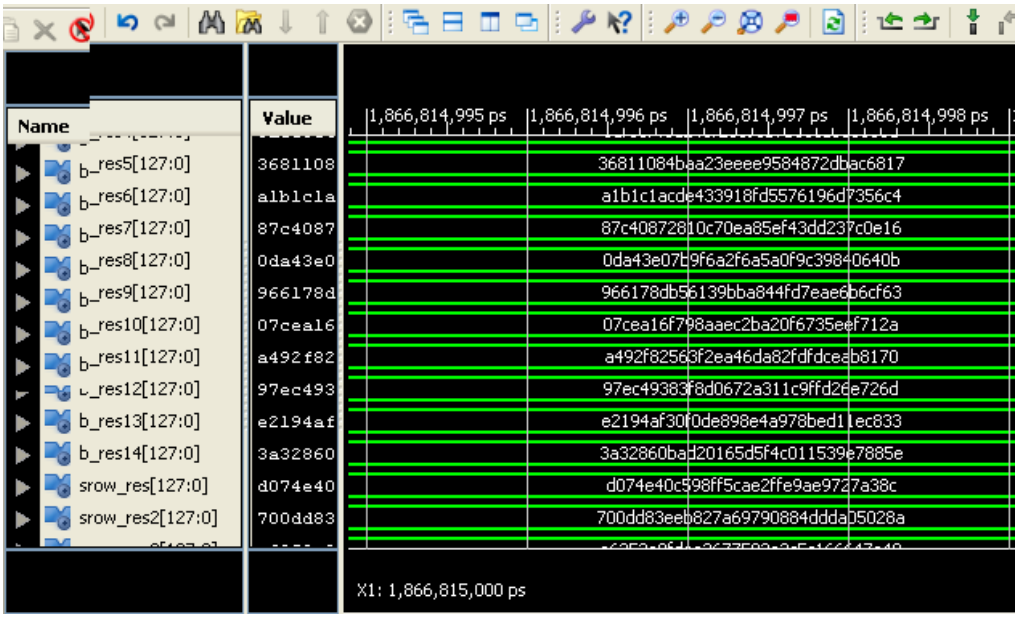

Figure 6.46 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. c

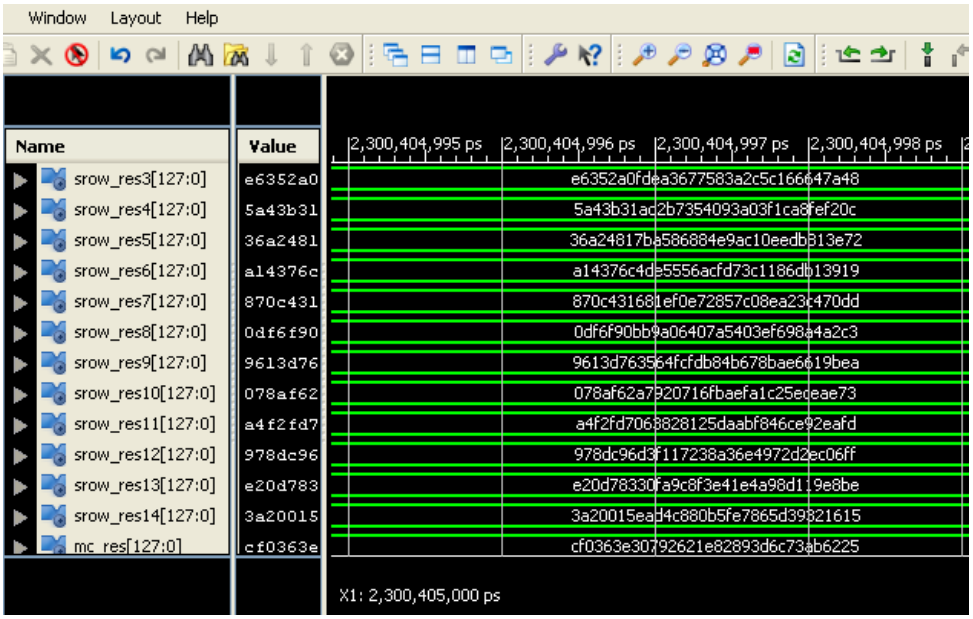

Figure 6.47 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. d

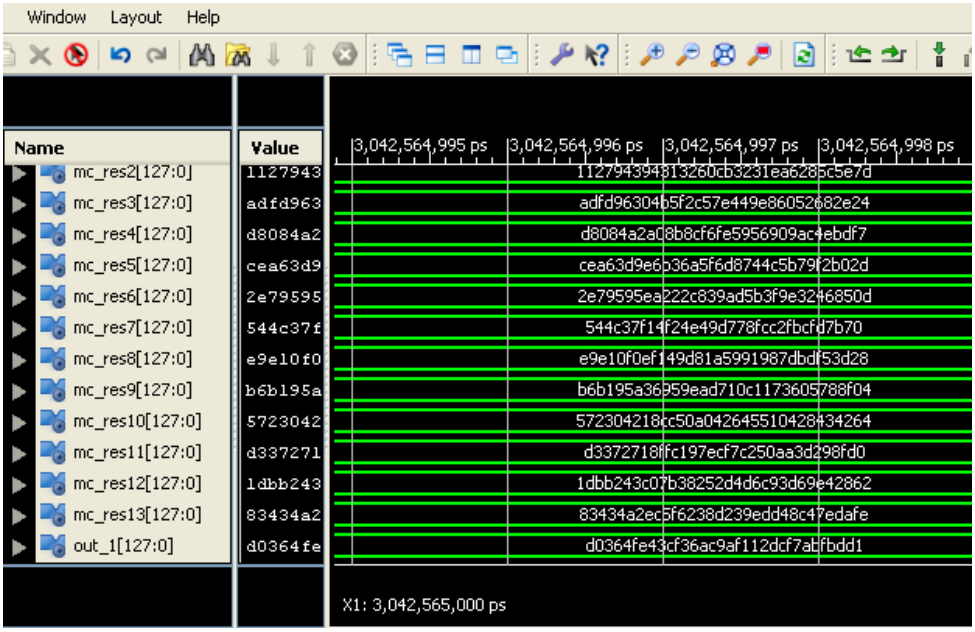

Figure 6.48 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. e

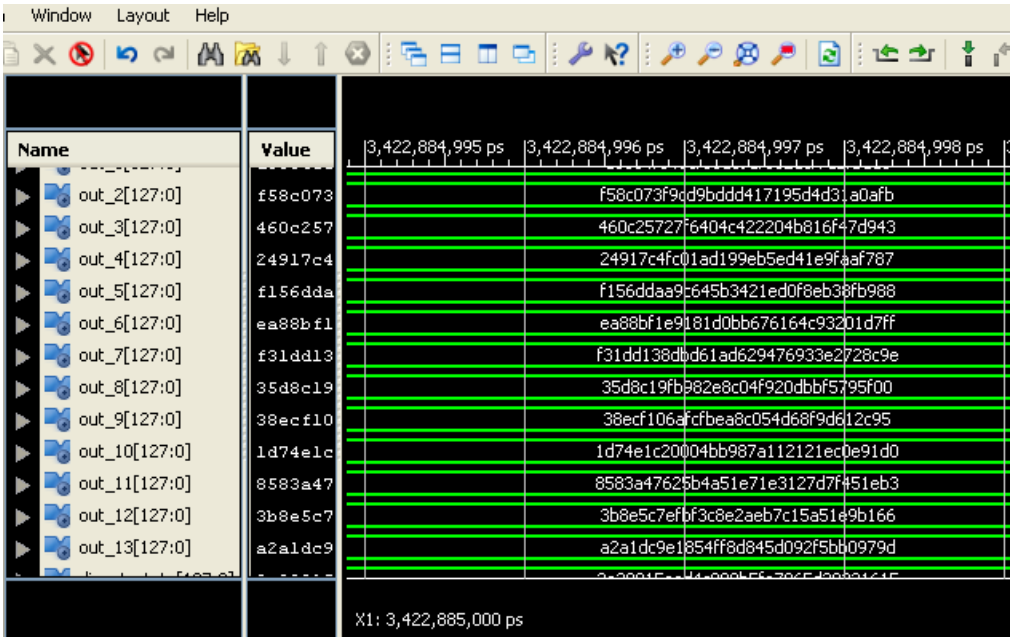

Figure 6.49 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. f

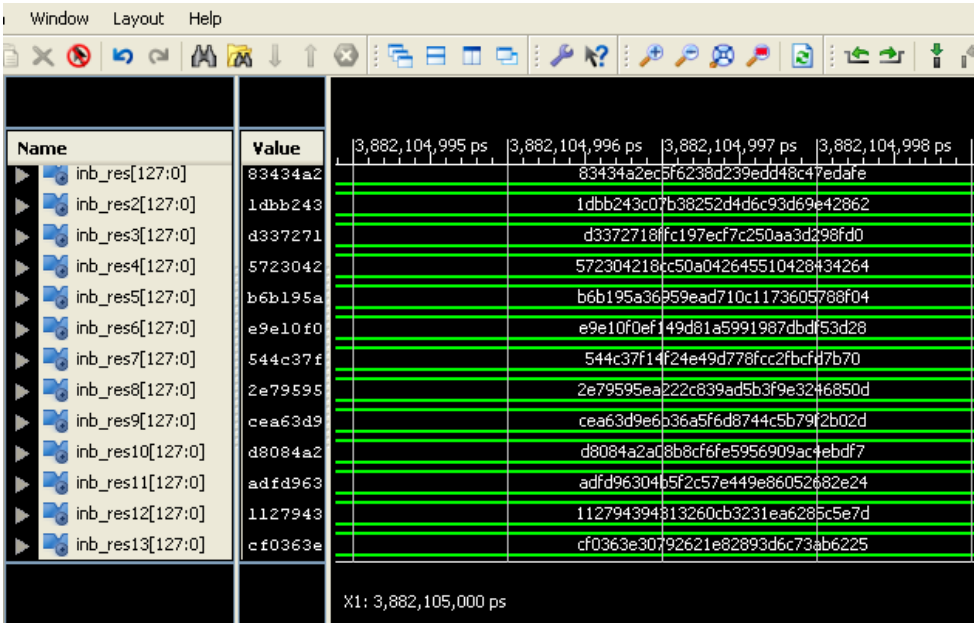

Figure 6.50 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. g

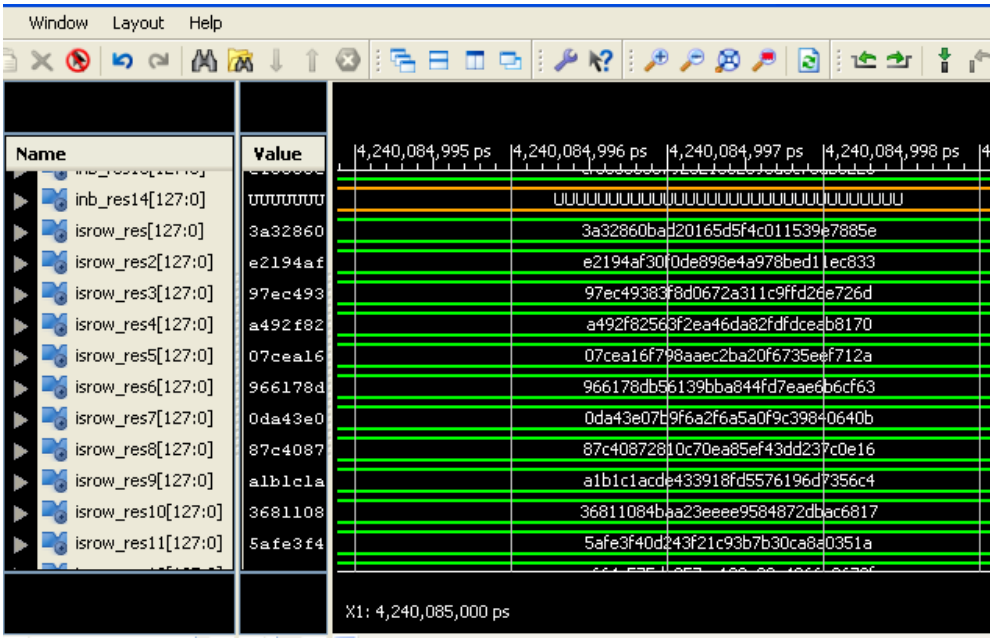

Figure 6.51 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. h

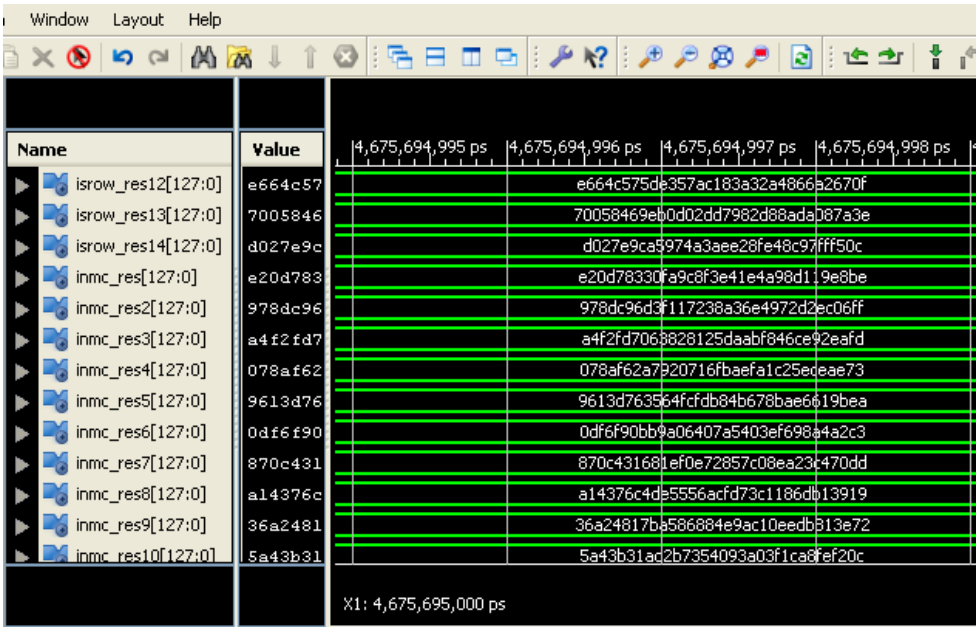

Figure 6.52 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. i

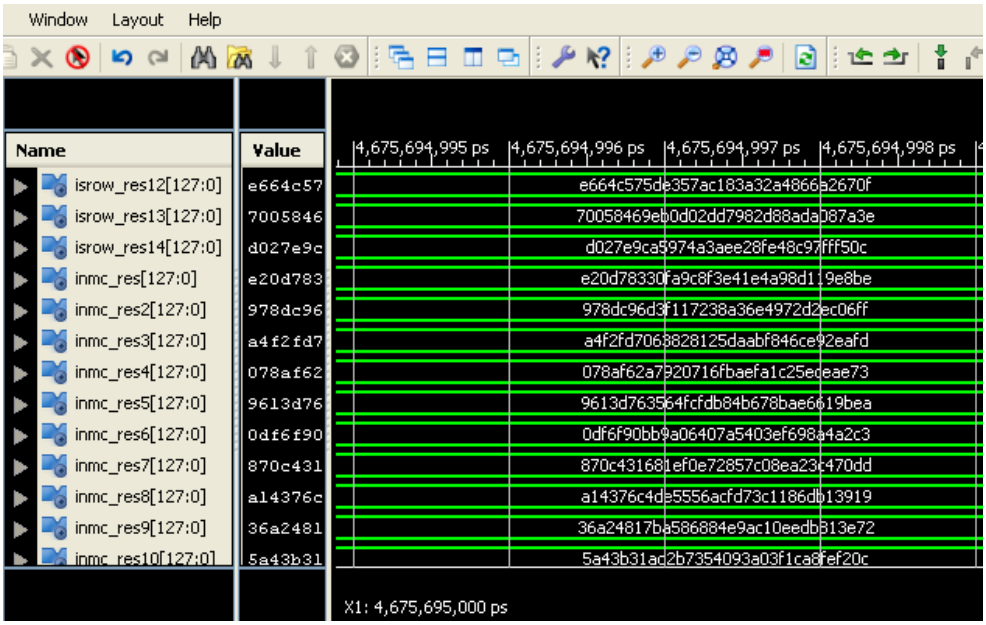

Figure 6.53 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. j

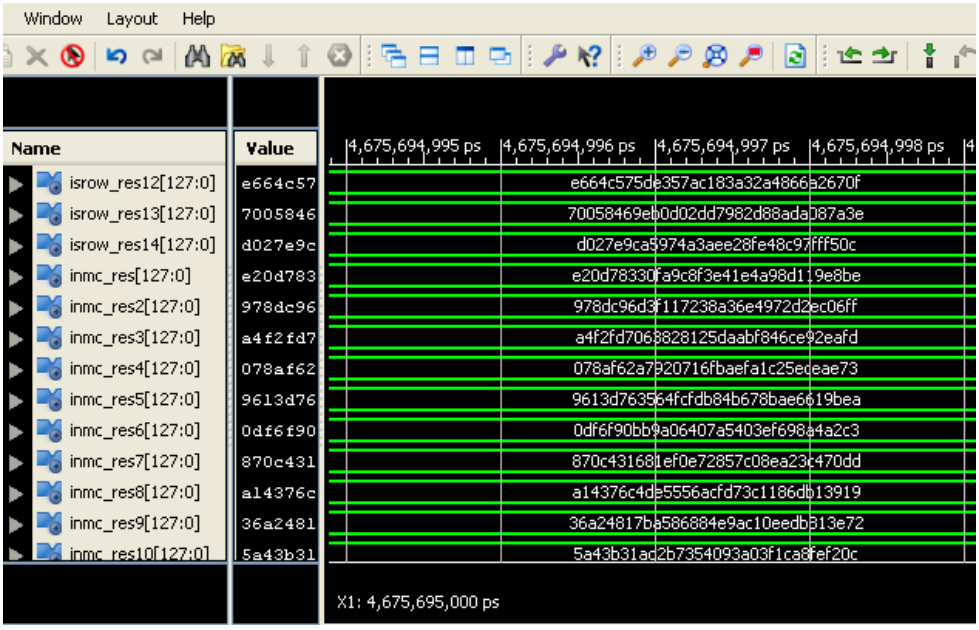

Figure 6.54 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. k

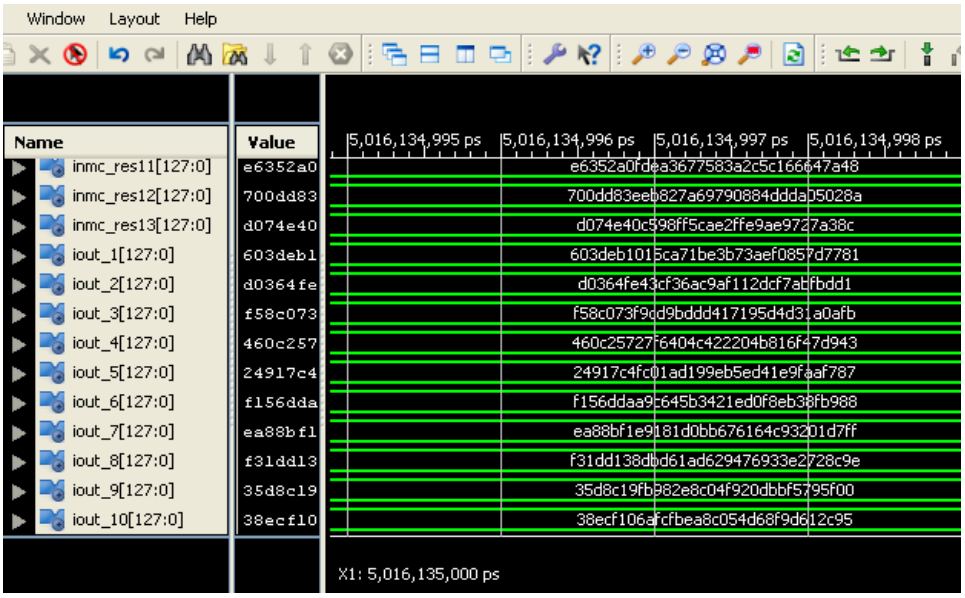

Figure 6.55 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. l

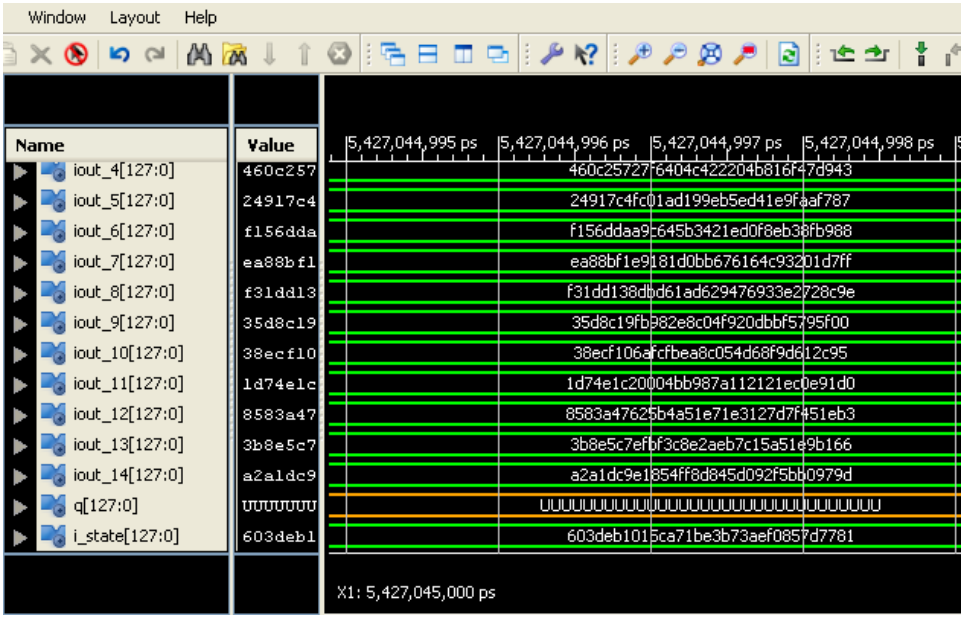

Figure 6.56 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. m

Simulation of AES with 256 Bits Security Key, input data a single 1 bit at Last point, displaying all intermediate processed data of round keys and intermediate transformations data

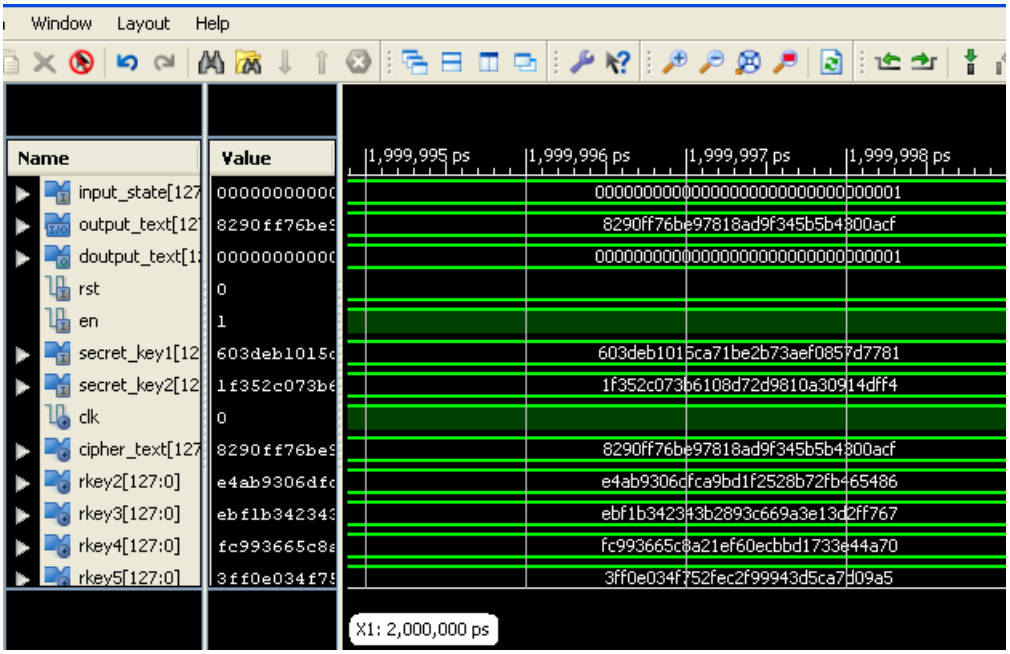

Figure 6.57 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No.i

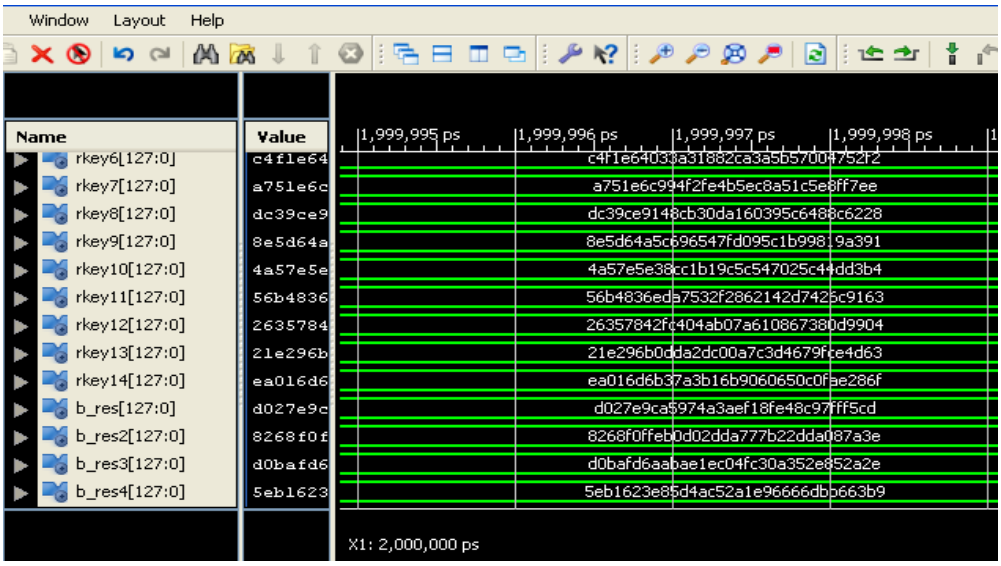

Figure 6.58 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. ii)

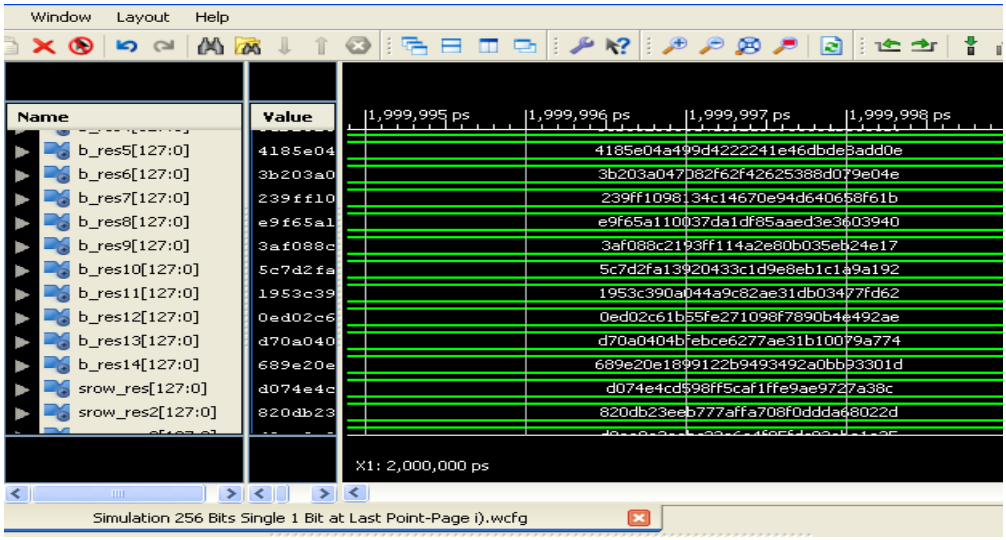

Figure 6.59 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. iii)

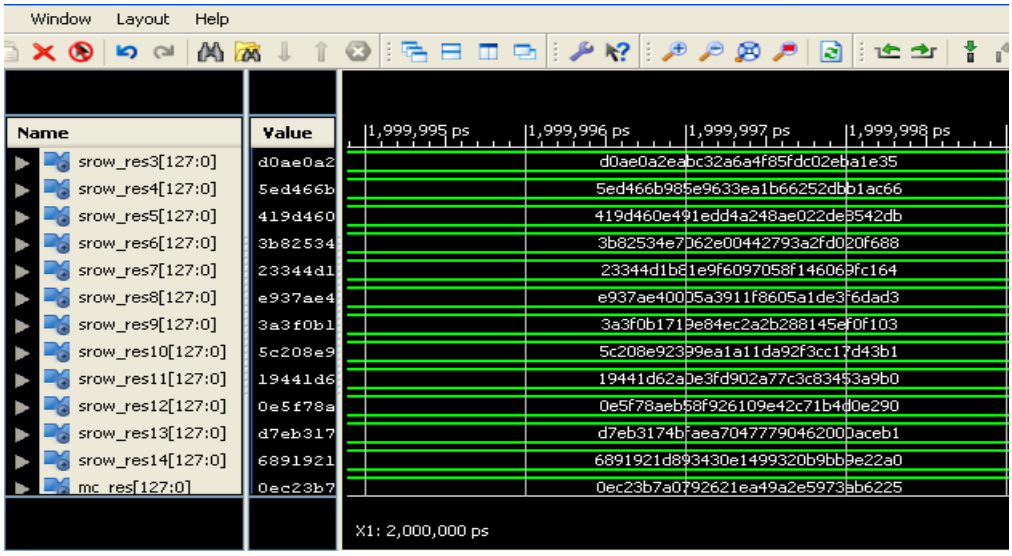

Figure 6.60 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. iv):

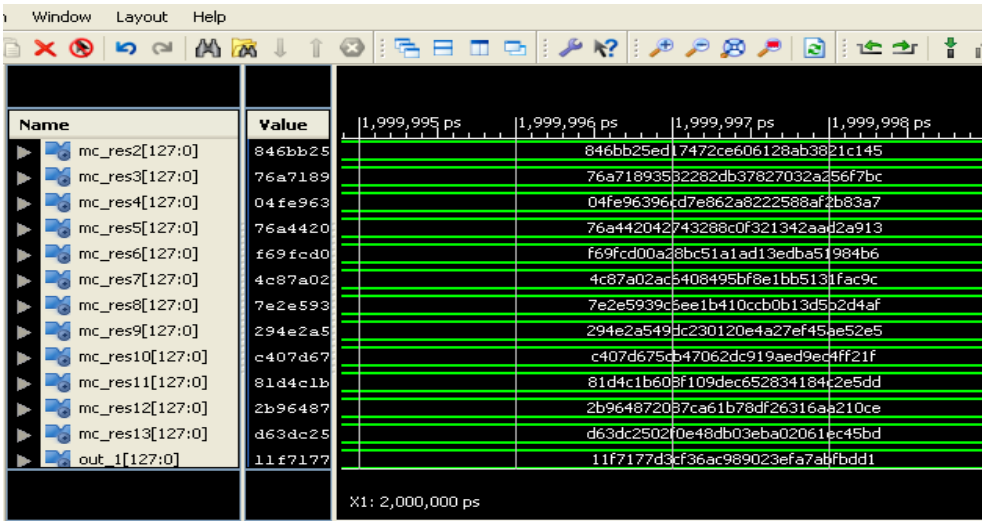

Figure 6.61 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. v):

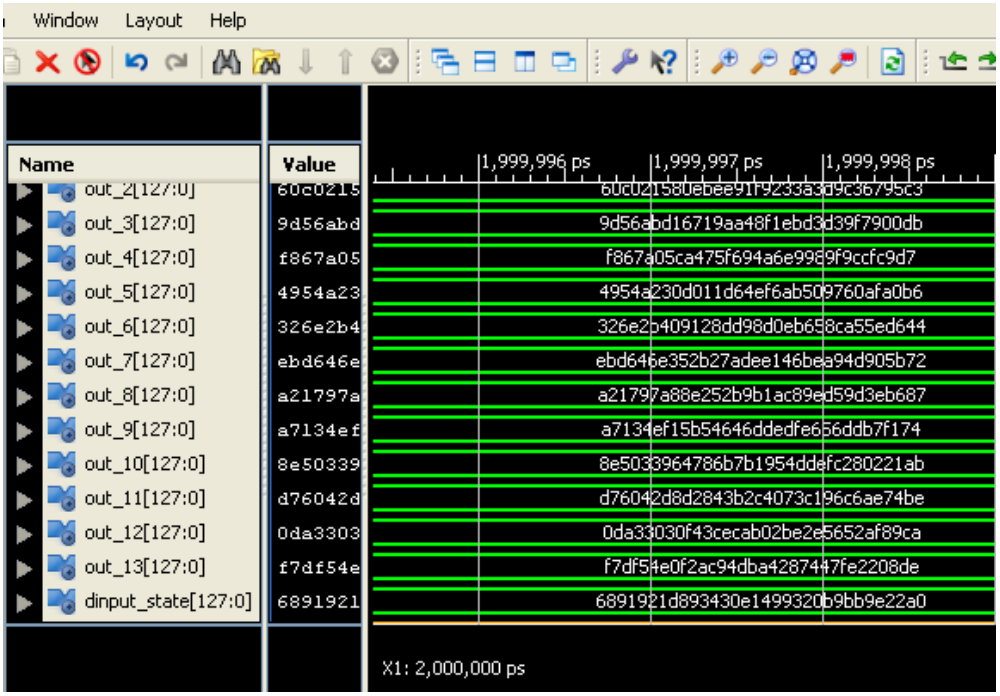

Figure 6.62 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. VI

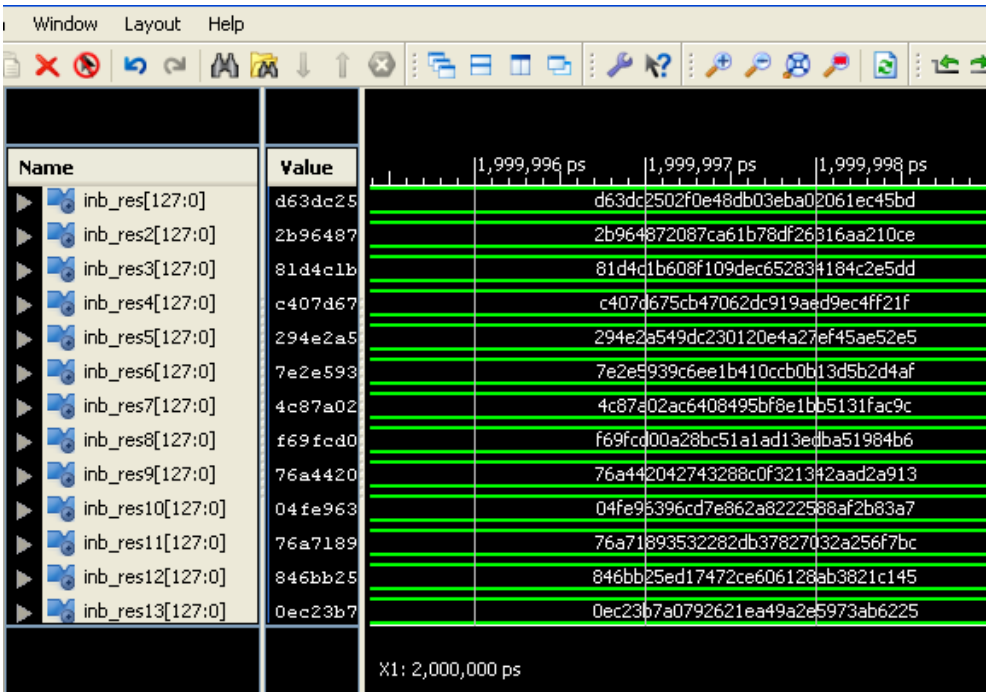

Figure 6.63 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. VII

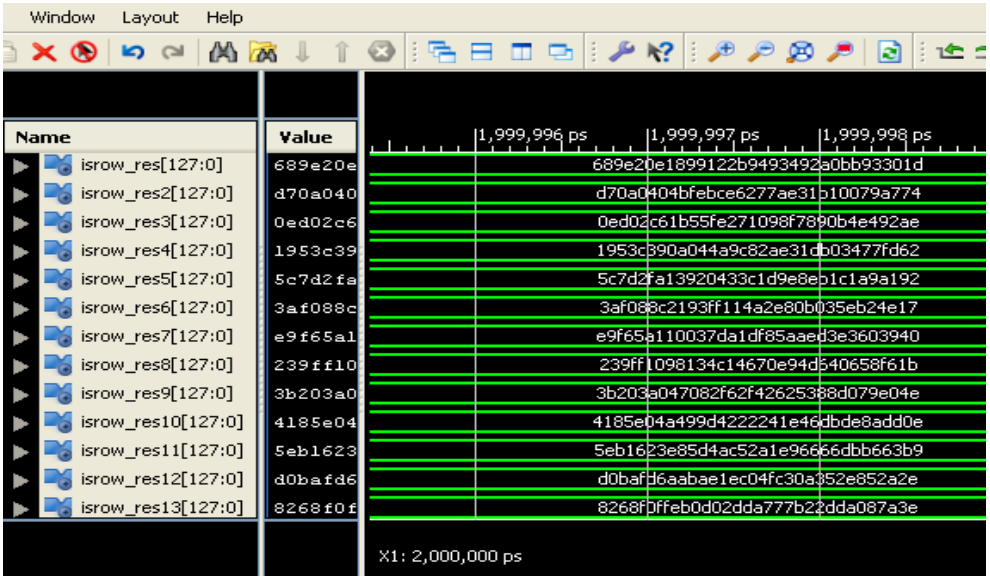

Figure 6.64 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. VIII

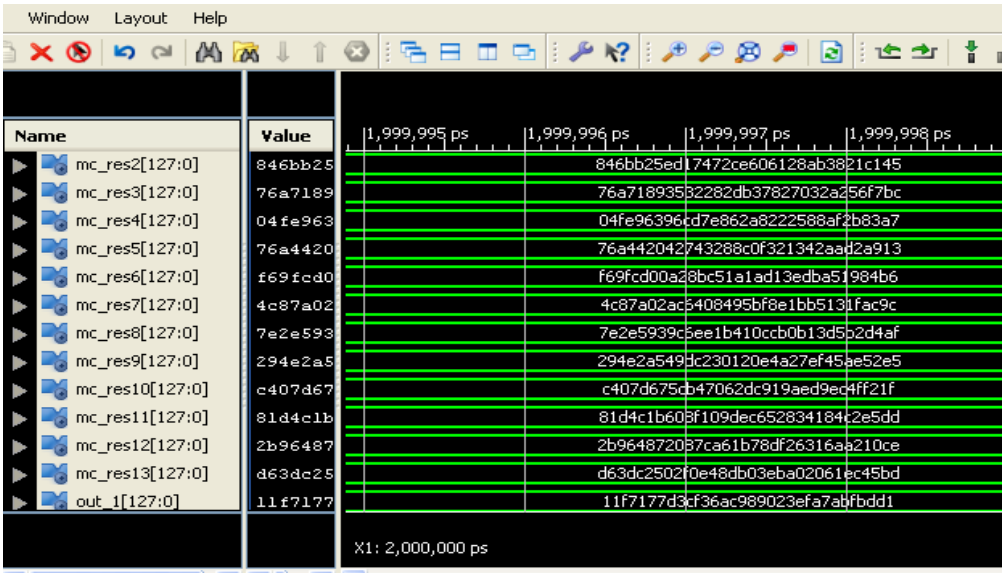

Figure 6.65 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. IX

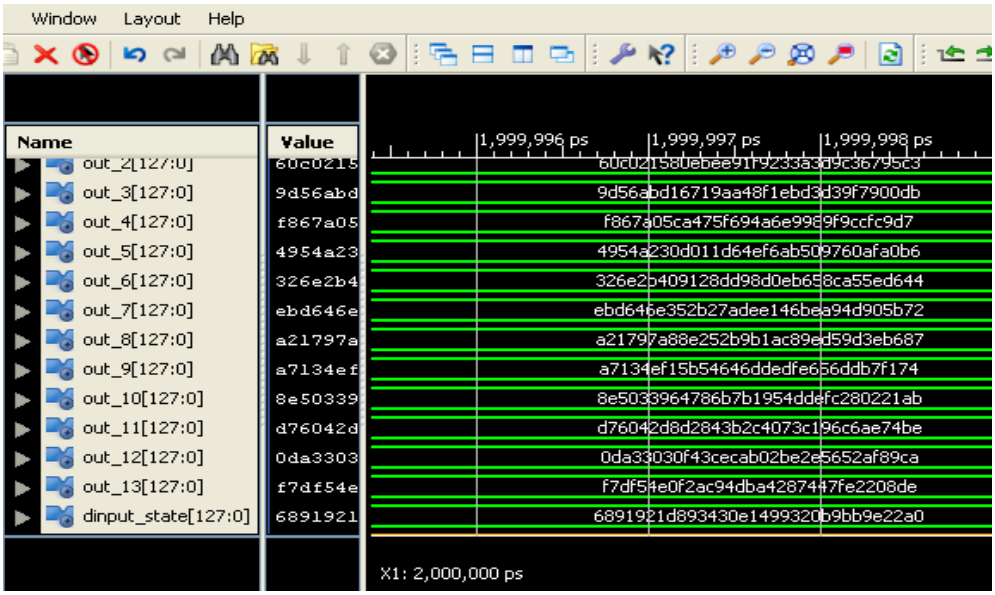

Figure 6.66 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. X

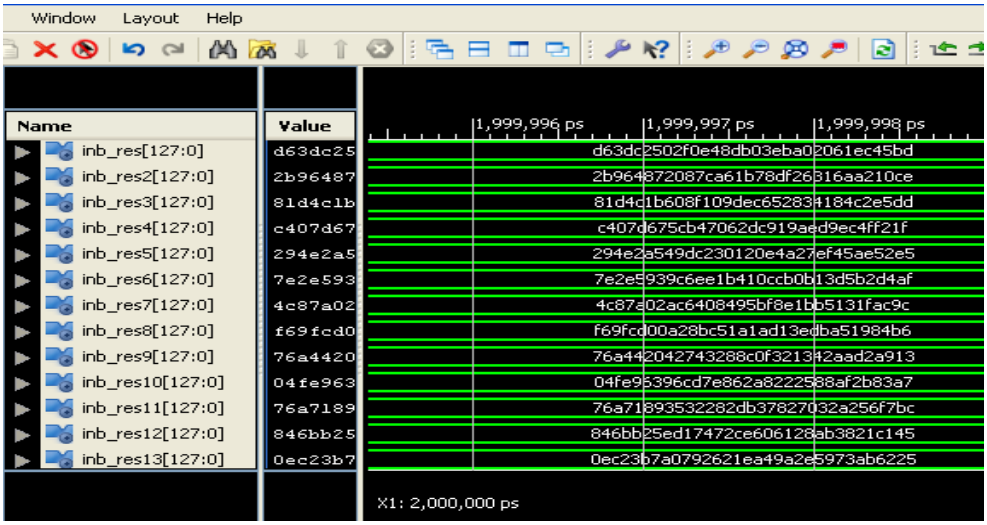

Figure 6.67 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. XI

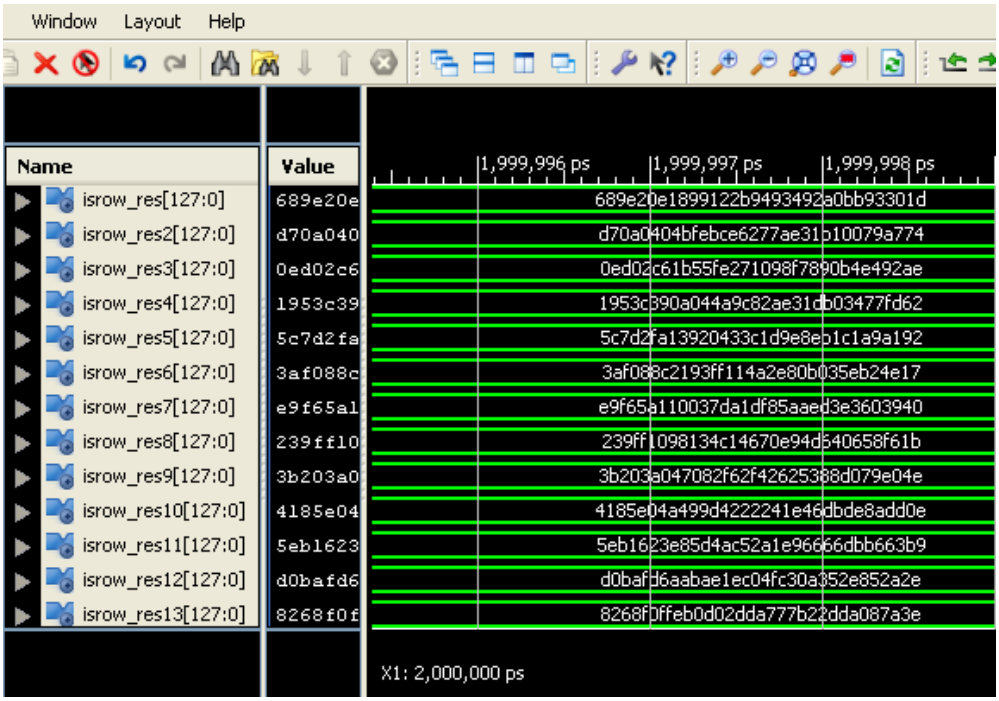

Figure 6.68 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. XII

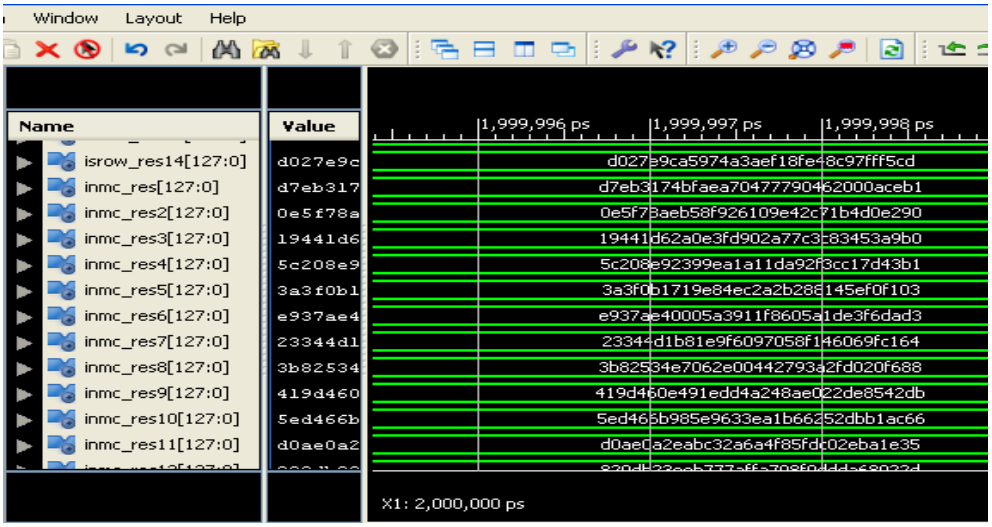

Figure 6.69 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. XIII

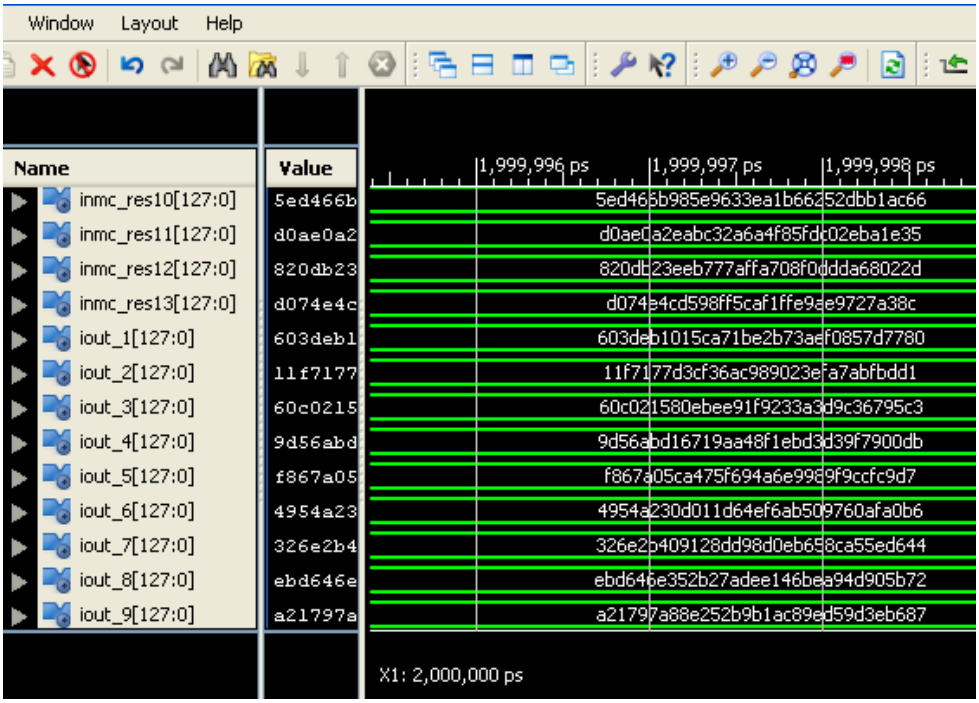

Figure 6.70 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. XIV

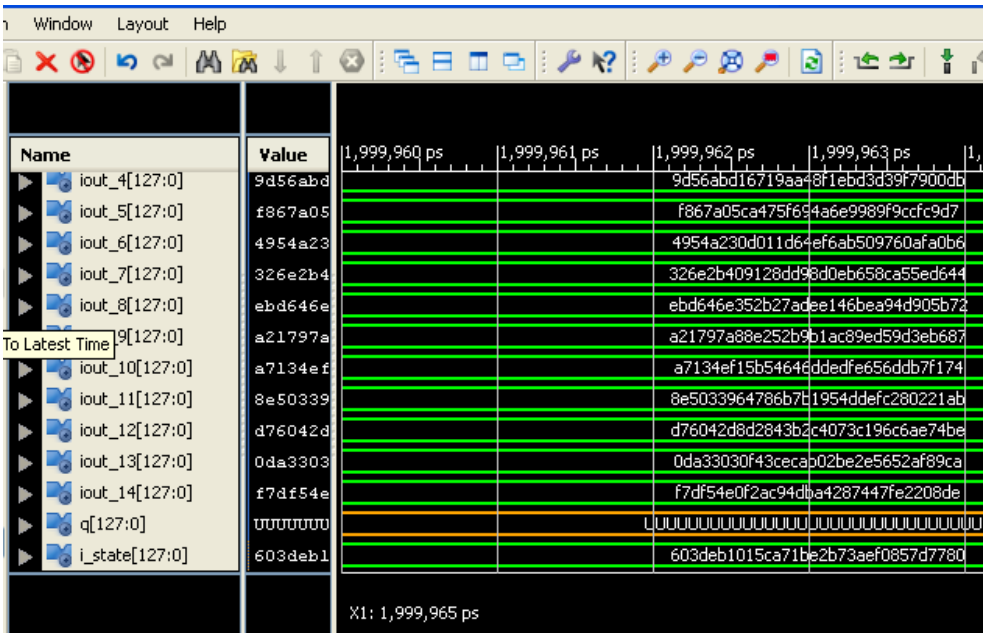

Figure 6.71 Simulation of AES with 256 bits Security Key, input data a Single bit at start Point sheet No. XV)

SYNTHESIS REPORT OF AES WITH 128 BITS SECURITY KEY AND 128 BITS DATA

Read as (HDL Synthesis Report: Macro Statistics)

# ROMs:	360;	256x8-bit ROM:	360;
# Registers:	380;	128-bit register	60;
8-bit register:	320;	# Xors	11901;
1-bit xor2:	9044;	1-bit xor3:	531;
1-bit xor4:	2304;	128-bit xor2:	22;

Advanced HDL Synthesis Report:

Read as (HDL Synthesis Report: Macro Statistics)

# ROMs:	360;	256x8-bit ROM:	360;
# Registers:	10240;	Flip-Flops:	10240;
# Xors:	11901;	1-bit xor2:	9044;
1-bit xor3:	531;	1-bit xor4:	2304;
128-bit xor2:	22;		

Final Register Report: Macro Statistics

# Registers:	10240;	Flip-Flops:	10240

Target Device: xc5vtx240t-2-ff1759

Total REAL time to Xst completion: 1.00 secs

Total CPU time to Xst completion: 0.63 secs

No partitions were found in this design.

Read as (Design Statistics: Cell Usage)

# IOs:	515	# BELS:	28646
# LUT2:	556	# LUT3:	451
# LUT4:	1049	# LUT5:	2269
# LUT6:	15649	# MUXF7:	5792
# MUXF8:	2880	# Flip-Flops /Latches:	10240
# FDC:	10240	# Clock Buffers:	2
# BUFGP:	2	# IO Buffers:	512
# IBUF:	256	# OBUF:	256

Device utilization summary: Selected Device: 5vtx240tff1759-2

Slice Logic Utilization:

Number of Slice Registers:	10240	out of	149760	6%
Number of Slice LUTs:	19974	out of	149760	13%
Number used as Logic:	19974	out of	149760	13%

Slice Logic Distribution:

Number of LUT Flip Flop pairs used:	22462			
Number with an unused Flip Flop:	12222	out of	22462	54%
Number with an unused LUT:	2488	out of	22462	11%
Number of fully used LUT-FF pairs:	7752	out of	22462	34%
Number of unique control sets:	1			

IO Utilization:

Number of IOs:	515			
Number of bonded IOBs:	514	out of	680	75%

Specific Feature Utilization:

Number of BUFG/BUFGCTRLs:	2	out of	32	6%

No Partitions were found in this design.

Timing report

Note: these timing numbers are only a synthesis estimate. For accurate timing information, please refer to the trace report

Generated after PLACE-and-ROUTE

Clock Information:

Clock Signal	Clock buffer(FF name)	Load
clk	BUFGP	10240

Asynchronous Control Signals Information:

Control Signal	Buffer(FF name)	Load
rst	BUFGP	10240

Timing Summary: Speed Grade: -2

Minimum period: 2.115ns (Maximum Frequency: 472.824MHz)

Minimum input arrival time before clock: 24.580ns

Maximum output required time after clock: 2.830ns

Maximum combinational path delay: No path found

Timing Detail: All values displayed in nanoseconds (ns)

Timing constraint: Default period analysis for Clock 'clk'

Clock period: 2.115ns (frequency: 472.824MHz)

Total number of paths / destination ports: 97984 / 10112

Delay: 2.115ns (Levels of Logic = 2)

Source: d171/output_90 (FF)

Destination: d172/DATAOUT_76 (FF)

Source Clock: clk rising; Destination Clock: clk rising;

Data Path: d171/output_90 to d172/DATAOUT_76 Gate Net

Read as (Cell:in->out fanout Delay Delay Logical Name (Net Name))

FDC:C->Q 11 0.396 0.947 d171/output_90 (d171/output_90)

LUT6:I0->O 1 0.086 0.600 d172/DATAOUT_xor0051127 (d172/DATAOUT_xor0051127)

LUT4:I1->O 1 0.086 0.000 d172/DATAOUT_xor0051144 (d172/DATAOUT_xor0051)

FDC:D -0.022 d172/DATAOUT_76

Total 2.115ns (0.568ns logic, 1.547ns route); (26.9% logic, 73.1% route)

Total REAL time to Xst completion: 273.00 secs

Total CPU time to Xst completion: 272.92 secs

Total memory usage is 352212 kilobytes

Number of errors : 0 (0 filtered)

Number of warnings: 30 (0 filtered)

Number of infos : 0 (0 filtered)

SYNTHESIS REPORT OF AES WITH 256 BITS SECURITY KEY

HDL Synthesis Report: Macro Statistics

ROMs: 500;

256x8-bit ROM: 500;

Registers: 532;

128-bit register: 84;

8-bit register: 448;

Xors: 18594;

1-bit xor2: 14469;

1-bit xor3: 767;

1-bit xor4: 3328;

128-bit xor2: 30;

1-bit xor3: 767;

1-bit xor4: 3328;

128-bit xor2: 30;

Final Register Report: Macro Statistics

Registers: 14336;

Flip-Flops: 14336;

Final Results:

RTL Top Level Output File Name: top_level.ngr; Top Level Output File Name: top_level

Output Format: NGC;

Optimization Goal: Speed;

Keep Hierarchy: No

Design Statistics: # IOs : 643

Cell Usage :

# BELS	: 39531	; #	LUT2	: 977	
# LUT3	: 738	; #	LUT4	: 1791	
# LUT5	: 3016	; #	LUT6	: 20995	
# MUXF7	: 8013	; #	MUXF8	: 4000	
# VCC	: 1	; # Flip-flops/Latches		: 14336	
# FDC	: 14336	; # Clock Buffers		: 2	
# BUFGP	: 2	; # IO Buffers		: 640	
# IBUF	: 384	; #	OBUF	: 256	

Device utilization summary: Selected Device: 5vtx240tff1759-2

Slice Logic Utilization:

Number of Slice Registers: 14336 out of 149760 9%

Number of Slice LUTs: 27517 out of 149760 18%

Number used as Logic: 27517 out of 149760 18%

Slice Logic Distribution:

Number of LUT Flip Flop pairs used: 31033

Number with an unused Flip Flop: 16697 out of 31033 53%

Number with an unused LUT: 3516 out of 31033 11%

Number of fully used LUT-FF pairs: 10820 out of 31033 34%

Number of unique control sets: 1

IO Utilization: Number of IOs: 643

Number of bonded IOBs: 642 out of 680 94%

Specific Feature Utilization: Number of BUFG/BUFGCTRLs: 2 out of 32 6%

Partition Resource Summary:

No Partitions were found in this design.

Clock Signal | Clock buffer(FF name) | Load |

clk | BUFGP | 14336 |

Asynchronous Control Signals Information:

Control Signal | Buffer(FF name) | Load |

---

rst | BUFGP | 14336 |

Timing Summary: Speed Grade: -2

Minimum period: 2.115ns (Maximum Frequency: 472.824MHz)

Minimum input arrival time before clock: 30.776ns

Maximum output required time after clock: 2.830ns

Maximum combinational path delay: No path found

Timing Detail:

All values displayed in nanoseconds (ns)

Timing constraint: Default period analysis for Clock 'clk'

Clock period: 2.115ns (frequency: 472.824MHz)

Total number of paths / destination ports: 138187 / 14208

Delay: 2.115ns (Levels of Logic = 2)

Source: d247/output_90 (FF)

Destination: d248/DATAOUT_76 (FF)

Source Clock: clk rising

Destination Clock: clk rising

Data Path: d247/output_90 to d248/DATAOUT_76

Total CPU time to Xst completion: 765.41 secs

Total memory usage is 434220 kilobytes

Number of errors : 0 (0 filtered)

Number of warnings : 9 (0 filtered)

Number of infos : 0 (0 filtered)

Table 6.2 Comparison of Synthesis Reports of 128 Bit Key System and 256 Bit Key Systems

S. no.	FPGA Parameter Requirements for the System	AES System with 128 Bit Security Key	AES System with 256 Bit Security key	Remarks
1.	No. of Flip Flops used	10240	14336	
2.	No. of ROMs used	360	500	
3.	Memory used	352.2 MBytes	434.2 MBytes	
4.	No. of LUT Slices used	19974	27517	
5.	Max. clock Freq.	472.82	472.82	Same
6.	Throughput	64 GBPS	64 GBPS	
7.	Processing Delay	2.115ns	2.115ns	Same
8.	No. of I/O Pins used	514	642	
9.	FPGA used	**xc5vtx240t-2-ff1759**	**xc5vtx240t-2-ff1759**	Same

SECURITY EVALUATION OF AES-256 BIT SECURITY KEY

A secure Block Cipher does not provide security on its own; it needs a secure system with secure protocol. The secure protocol require GCM type mode of operation. The encryption scheme requires a Block Cipher with long security key of 256 bit for ensuring high level of security. AES-256 is still considered a secured against attacks using quantum cryptanalysis. The US government is still considered AES with security key for the transmission of TOP SECRET information for protection of their secrets.

The AES-256 algorithm itself requires a well-protected secret key and secure implementation for the protection against side channel attacks, it must be ensured as FIP compliant. As far I know from Google search AES-256 Block Cipher has not been broken, the most ciphers cannot be proven to be secure. In information theoretical sense, only one time pad type algorithm may be secure.

The attacks are indeed possible and they reduce the strength of AES *for specific use cases* to a value that. Basically, we should not use AES-256 to build a hash function.

Table 6.3. Best attacks on AES-256

Attack	Rounds	Keys	Data	Time	Memory	Source
Related –key boomerang	14	256	4	$2^{99.5}$	2^{77}	Sec.5
Partial sums	9	256	2^{85}	2^{226}	2^{32}	[11]
Related - key rectangle	10	64	2^{114}	2^{173}	?	[6, 14]
	14	2^{35}	2^{131}	2^{171}	2^{65}	[7]

The differential trails for the attacks are based on the idea of finding local collisions in the block cipher. The optimal key-schedule trails should be based on low-weight code words in the key schedule. Boomerang-switching techniques are exploited to gain free rounds in the middle of the cipher. The related – key boomerang attacks are still mainly of theoretical interest and do not present a threat to practical applications using AES.

Researchers (Bogdanov, A., & Rechberger, C. 2010) and (Bogdanov, A., Lauridsen, M. M., & Tischhauser, E. 2014) worked on novel techniques of block cipher Biclique cryptanalysis of the full AES, which leads to the following results, using the first recovery method. They have tried the pre-image search for compression functions based on the full AES versions faster than brute force. Some Results on AES (BKR11) of Key Recovery:

Table 6.4 AES- 128 Secret Key Recovery

Rounds	Data	Computations	Memory	Biclique Length in Rounds
8	$2^{126.33}$	$2^{124.97}$	2^{102}	5
8	2^{127}	$2^{125.64}$	2^{32}	5
8	2^{88}	$2^{125.34}$	2^{8}	3
10	2^{88}	$2^{126.18}$	2^{8}	3

Table 6.5 AES – 192 Secret Key Recoveries:

Rounds	Data	Computations	Memory	Biclique Length in Rounds
9	2^{80}	$2^{188.8}$	2^{8}	4
12	2^{80}	$2^{189.74}$	2^{8}	4

Table 6.6 AES – 256 Secret Key Recoveries:

Rounds	Data	Computations	Memory	Biclique Length in Rounds
9	2^{120}	$2^{253.1}$	2^{8}	6
9	2^{120}	$2^{251.92}$	2^{8}	4
14	2^{40}	$2^{254.42}$	2^{8}	4

RELATED KEY ATTACK ON AES-256

AES-256 cannot model an ideal Cipher in theoretical construction, a related key distinguisher with one out of every235 keys, 2^{120} data, and negligible memory but with time complexity. The distinguisher carries out a key recovery attack with complexity of 2^{131} time and 2^{65} memories. The differential weakness in the key schedule of AES-256 such as slow diffusion is identified. The identified

slow diffusion is matched with differential properties of the round function. The propagation patterns between two round difference of AES and thus generate local collisions for AES. It is possible to find out the low weight difference in the sub keys and zero difference in the 128 bit block.

The four local collisions can be concatenated together and add another 6-round trail on top to make full 14 rounds of the AES-256 algorithm. There are 41 active S-boxes, 5 in the key schedule and 36 in the block. The triangulation algorithm tools are applied to find out collisions in the hash function, for finding keys and plaintexts in order to conform to the trail.

In related Key attacks, by changing the top two rounds of the trails, one obtain a differential trail with only 24 active S-Boxes in total, 19 in the round function and 5 in the key schedule. The differential distinguisher for AES-256 can be generated from the above trail which works for one key out of 2^{35} and has complexity of 2^{120} data and time but with negligible memory. The derived distinguisher may be used for a key recover attack on AES-256 with total complexity of 2^{35+96} time and 2^{65} memories.

LOCAL COLLISIONS IN AES

The researchers Chabaud and Joux suggested injecting a difference into the internal state thereby causing a disturbance. Then the correction is to be done with the next injection. The resulting difference pattern is spread out due to message schedule causing more disturbances in other rounds. The aim is to have a few disturbances in order to keep low complexity of the attack. The attacker cannot control the key thus the attack should work for any key pair with a given difference.

There is a one active S-box in the internal state, and five non –zero byte differences in the two consecutive sub-keys. This differential holds with probability 2^{-6} if we use an optimal differential for an S-box.

Figure 6.69 Local Collisions in AES

Due to the key schedule the differences spread to the other sub-keys thus forming the key schedule difference. The resulting key schedule difference can be viewed as a set of local collisions, which cause expansion of the disturbance, called also as disturbance vector; the correction differences compensate each other. The probability of the full differential trail is then determined by the number of active S-boxes.

Table 6.6 Best Attacks on AES-256:

S No.	Types of Attacks	No. of Rounds	No. of Keys	Data	Time	Memory
1.	Known key integral Partial sums Related Key rectangles	7 9 10	1 256 64	2^{56} 2^{85} 2^{114}	2^{56} 2^{226} 2^{173}	2^{56} 2^{32}
2.	q- multi-collisions	14	2q	2q	$q.2^{67}$	-
3.	Partial multi-collisions	14	2q	2q	$q.2^{37}$	-
4.	Related Key Distinguisher	14	2^{35}	2^{119}	2^{119}	-
5.	Related-key Key Recovery	14	2^{35}	2^{96}	2^{96}	2^{65}

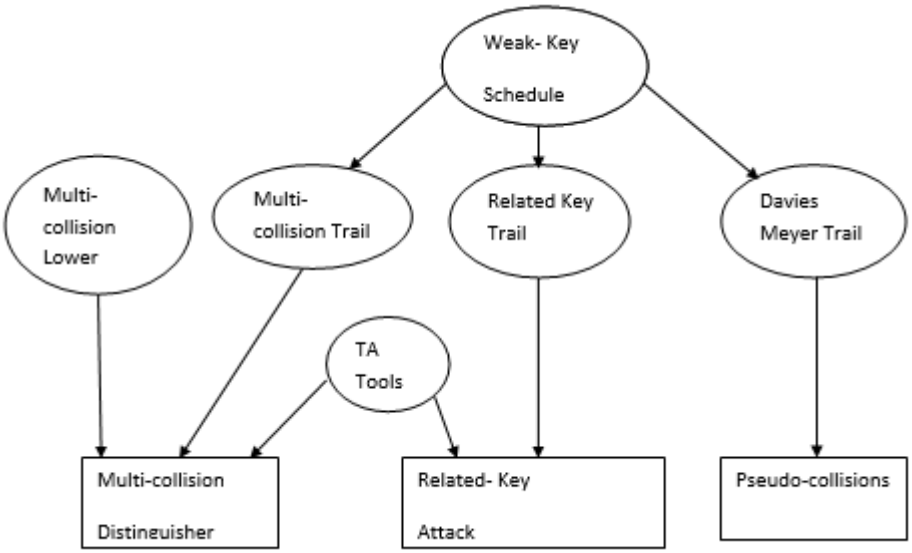

Figure 6.70 Related Key Attack Scheme for AES

Chapter 7

CONCLUSION AND FUTURE RESEARCH WORK

CHAPTER 7 – CONCLUSION AND FUTURE RESEARCH WORK

The comparison of various open source intrusion detection systems, industrial intrusion prevention systems for cyber security threats prevention and adopting planned security strategy of the industrial infrastructure have been discussed and proposed. Distribution network protocol architectures, analyzer design philosophy, parser designing, implementation with security performance evaluation has been discussed to ensure data security for modern global SCADA systems. IPSec is used for providing private secure communications over Internet Protocol (IP) protocols, identifying various encryption and authentication algorithms. The performance, efficiency and high level security in IPSec based Mobile VPN by selecting appropriate components of VPN Tunnel discussed.

The data encryption implementation schemes for AES algorithm with security key of 128 Bits and 256-bit security key has been proposed for FPGA chip based systems. The top-level entity design system has been designed, software program has been developed, and simulation results generated, comparison tables for its performance with earlier researcher has been made. The implementation of AES algorithm for processing data for encryption with security key of 256 bits has been implemented using FPGA chip no. XC5vtx240t-2-ff1759, and tests performed with different sets of input data. The simulation results of processing data and generation at intermediate transformations of data processing was generated and found correct values of ciphered data and generation of original data at receiver output. Synthesis Reports of chip design for the FPGA chip no. XC5vtx240t-2-ff1759 has been generated and attached. Comparison Table of our chip design and earlier researcher show improvements in design philosophy.

This proposed system FPGA chip implementation requires 515 input and output ports. The requirement of input and output ports is very large, which can be reduced considerably by using internal serial to parallel registers for input security key and input data respectively, and parallel to serial register for output

data inside FPGA device to reduce pin count from 384 to 3 for I/O ports. An attempt has been made for designing highly secure AES Implementation on FPGA with long size key for data transmission between Server system and other connected corporate business computers for Petroleum Industry and other Industries.

The performance of the proposed, implemented cipher transmitter system and receiver deciphering system was checked with different input data. The cipher security by transmitting just single one-bit pulse among 128 bits input data has been ensured and confirmed as per simulation results shown in the Chapter No. 6. The just transmitting single bit data has been verified for both of two proposed and implemented FPGA system as shown in simulation results. The single bit data may be as a starting bit, at the last bit position or at any in between position ciphering security is completely ensured and verified, as shown in simulation results.

The new method for generation of individual round keys from the given security key of 256 bits of AES have been proposed, adopted and analyzed for implementation, for increasing the speed of processing data coding. The Notations and Notions have been proposed and then calculated the every individual round key from the given security key for AES implementation in FPGA applications. After all the round keys are generated, these may be stored till the given code is in use. Immediately the individual round key is generated, it can be used for processing the data for that specific round, rather than waiting for generation of all round keys, this is the novelty of this proposed technique of round generation. This scheme of round key generation will help in increasing the speed of data processing in some applications where sharp response is needed.

The AES algorithm implementation has all linear operation except Substitution Box (S-Box) which is a non linear operation and there is a scope of optimization of its speed of operation and reduction of silicon chip area. The look up table (LUT) implementation has an unbreakable delay to pick up value of S-Box from PROM stored chip, for its implementation. The Combinational Logic Circuits

Implementation for S-Box has the limitation of not very fast in speed and size of circuit not small. Another scheme for S-Box implementation is Composite Field Architecture (CFA), which may be designed to have a small in size, and fast in speed for high throughput applications since it can be optimized for better architecture and algorithmic operation. The multiplicative Inversion technique by using isomorphic mapping with common sub expression elimination in sub field helps in reducing chip area. FPGA Implementation using CFA technique is used in achieving high-speed data processing for some applications analyzed. The researchers have improved the performance of Cipher Systems by using multi-processing cores for parallel processing capabilities, and increasing in-build data security have proposed by researchers in recent research papers for high speed and secure AES Implementations.

The researchers have improved the performance of Ciphers Systems by using multi-processing cores for parallel processing capabilities and by proposing the new powerful processor Instructions set. Instruction Sets of AES-NI meant for increasing the speed of AES Implementation and increasing in-build data security have proposed by researchers in recent research papers for high speed and secure AES Implementations. These instructions have improved the performance in comparison of pure software implementations, having full flexibility of usability with all standard key lengths, standard mode of operations. These instructions have provided security enhancement also, by eliminating major timing and cache based attacks. These instructions are designed to carry out complex and computationally better steps of AES algorithm, which in turn accelerate the execution of AES algorithm. These Instructions improved the performance by a factor of 4 to 10 in comparison to complete software programs.

Authenticated Encryption with Associated Data (AEAD) Modes, EAX mode of Cipher Operation has been proposed by researchers to increase speed of implementation for multimedia applications. The scope of increasing ciphering speed of data processing by means of new specialized instruction sets for repeated operational steps in hardware, to accelerate the performance of Galois Field fixed field constant multiplication, an important element of AES

algorithm, in comparison to pure software implementation speed. Software optimization accelerator is always there for future researchers to undertake.

REFERENCES

Akkar, M. L., & Giraud, C. (2001, May). An implementation of DES and AES, secure against some attacks. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 309-318). Springer, Berlin, Heidelberg.

Alsiherov, F., & Kim, T. (2010, May). Secure SCADA network technology and methods. In Proceedings of the Twelfth WSEAS International Conference on Automatic Control, Modeling and Simulation (pp. 434-438).

Anand, A., & Patel, B. (2012). An overview on intrusion detection system and types of attacks it can detect considering different protocols. International Journal of Advanced Research in Computer Science and Software Engineering, 2(8).

Anderson, R., & Kuhn, M. (1996, November). Tamper resistance-a cautionary note. In Proceedings of the second Usenix workshop on electronic commerce (Vol. 2, pp. 1-11).

Anderson, R., & Kuhn, M. (1997, April). Low cost attacks on tamper resistant devices. In International Workshop on Security Protocols (pp. 125-136). Springer, Berlin, Heidelberg.

Aoki, K., & Sasaki, Y. (2009, August). Meet-in-the-middle preimage attacks against reduced SHA-0 and SHA-1. In Annual International Cryptology Conference (pp. 70-89). Springer, Berlin, Heidelberg.

Aziz, A., & Ikram, N. (2007). An FPGA-based AES-CCM Crypto Core For IEEE 802.11 i Architecture. IJ Network Security, 5(2), 224-232.

Bahrak, B., & Aref, M. R. (2007). A novel impossible differential cryptanalysis of AES. In proceedings of the Western European Workshop on Research in Cryptology (Vol. 2007).

Bahrak, B., & Aref, M. R. (2008). Impossible differential attack on seven-round AES-128. IET Information Security, 2(2), 28-32.

Beggs, C. (2008). A holistic SCADA security standard for the Australian context.

Bellare, M., & Kohno, T. (2003, May). A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 491-506). Springer, Berlin, Heidelberg.

Bellare, M., Rogaway, P., & Wagner, D. (2004, February). The EAX mode of operation. In International Workshop on Fast Software Encryption (pp. 389-407). Springer, Berlin, Heidelberg.

Biham, E., Biryukov, A., & Shamir, A. (1999, March). Miss in the Middle Attacks on IDEA and Khufu. In International Workshop on Fast Software Encryption (pp. 124-138). Springer, Berlin, Heidelberg.

Biham, E., Dunkelman, O., & Keller, N. (2001, May). The rectangle attack—rectangling the Serpent. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 340-357). Springer, Berlin, Heidelberg.

Biham, E., Dunkelman, O., & Keller, N. (2002, February). New results on boomerang and rectangle attacks. In International Workshop on Fast Software Encryption (pp. 1-16). Springer, Berlin, Heidelberg.

Biham, E., Dunkelman, O., & Keller, N. (2005, May). Related-key boomerang and rectangle attacks. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 507-525). Springer, Berlin, Heidelberg.

Biryukov, A., & Khovratovich, D. (2009, December). Related-key cryptanalysis of the full AES-192 and AES-256. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 1-18). Springer, Berlin, Heidelberg.

Biryukov, A., Khovratovich, D., & Nikolic, I. (2009). Examples of differential multicollisions for 13 and 14 rounds of AES-256. IACR Cryptology ePrint Archive, 2009, 242.

Biryukov, A., Khovratovich, D., & Nikolić, I. (2009, August). Distinguisher and related-key attack on the full AES-256. In Annual International Cryptology Conference (pp. 231-249). Springer, Berlin, Heidelberg.

Biryukov, A., & Nikolić, I. (2010, May). Automatic search for related-key differential characteristics in byte-oriented block ciphers: application to AES, Camellia, Khazad and others. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 322-344). Springer, Berlin, Heidelberg.

Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., & Shamir, A. (2010, May). Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 299-319). Springer, Berlin, Heidelberg.

Blömer, J., Guajardo, J., & Krummel, V. (2004, August). Provably secure masking of AES. In International workshop on selected areas in cryptography (pp. 69-83). Springer, Berlin, Heidelberg.

Bogdanov, A., & Rechberger, C. (2010, August). A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN. In International Workshop on Selected Areas in Cryptography (pp. 229-240). Springer, Berlin, Heidelberg.

Bogdanov, A., Lauridsen, M. M., & Tischhauser, E. (2014). AES-Based Authenticated Encryption Modes in Parallel High-Performance Software. IACR Cryptology ePrint Archive, 2014, 186.

Bollapragada, V., Khalid, M., & Wainner, S. (2005). IPSec VPN Design. Cisco Press.

Bulens, P., Standaert, F. X., Quisquater, J. J., Pellegrin, P., & Rouvroy, G. (2008, June). Implementation of the AES-128 on Virtex-5 FPGAs. In International Conference on Cryptology in Africa (pp. 16-26). Springer, Berlin, Heidelberg.

Calomel.org (2015). AES-NI SSL Performance a study of AES-NI acceleration using OpenSSL, https://calomel.org/aesni_ssl_performance.html

Canright, D. (2005, August). A very compact S-box for AES. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 441-455). Springer, Berlin, Heidelberg.

Canright, D., & Batina, L. (2008, June). A very compact "perfectly masked" S-box for AES. In International Conference on Applied Cryptography and Network Security (pp. 446-459). Springer, Berlin, Heidelberg.

Centre for the Protection of National Infrastructure (CPNI), USA, (2006, November) Good Practices Guide for Process Control and SCADA Security. https://www.controlglobal.com/assets/Media/MediaManager/wp_06_niscc_scada.pdf

Chabaud, F., & Joux, A. (1998, August). Differential collisions in SHA-0. In Annual International Cryptology Conference (pp. 56-71). Springer, Berlin, Heidelberg.

Chari, S., Jutla, C. S., Rao, J. R., & Rohatgi, P. (1999, August). Towards sound approaches to counteract power-analysis attacks. In Annual International Cryptology Conference (pp. 398-412). Springer, Berlin, Heidelberg.

Chodowiec, P., & Gaj, K. (2003, September). Very compact FPGA implementation of the AES algorithm. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 319-333). Springer, Berlin, Heidelberg.

CISCO (2014). Cisco 2014 Annual Security Report, http://www.cisco.com.

Coates, G. M., Hopkinson, K. M., Graham, S. R., & Kurkowski, S. H. (2008). Collaborative, trust-based security mechanisms for a regional utility intranet. IEEE Transactions on power systems, 23(3), 831-844.

Coates, G. M., Hopkinson, K. M., Graham, S. R., & Kurkowski, S. H. (2009). A trust system architecture for SCADA network security. IEEE Transactions on Power Delivery, 25(1), 158-169.

Dan Ehrenreich, ShlomoLiberman,“ Motorola’s MDLC Protocol Enhances DNP 3.0 Based SCADA Systems”.

Demirci, H., & Selçuk, A. A. (2008, February). A meet-in-the-middle attack on 8-round AES. In International Workshop on Fast Software Encryption (pp. 116-126). Springer, Berlin, Heidelberg.

Dirci, H., Taşkın, İ., Çoban, M., & Baysal, A. (2009, December). Improved meet-in-the-middle attacks on AES. In International Conference on Cryptology in India (pp. 144-156). Springer, Berlin, Heidelberg.

Dunkelman, O., & Keller, N. (2010). The effects of the omission of last round's MixColumns on AES. Information Processing Letters, 110(8-9), 304-308.

Dunkelman, O., Keller, N., & Shamir, A. (2010, August). A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3G telephony. In Annual cryptology conference (pp. 393-410). Springer, Berlin, Heidelberg.

Dunkelman, O., Keller, N., & Shamir, A. (2010, December). Improved single-key attacks on 8-round AES-192 and AES-256. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 158-176). Springer, Berlin, Heidelberg.

Dunkelman, O., Sekar, G., & Preneel, B. (2007, December). Improved meet-in-the-middle attacks on reduced-round DES. In International Conference on Cryptology in India (pp. 86-100). Springer, Berlin, Heidelberg.

Elbayoumy, A. D., & Eldemerdash, H. Design and Implementation of Multi-Rate Encryption Unit Based on Customized AES.

Elbirt, A. J., Yip, W., Chetwynd, B., & Paar, C. (2001). An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 9(4), 545-557.

Fan, C. P., & Hwang, J. K. (2008). FPGA implementations of high throughput sequential and fully pipelined AES algorithm. International journal of electrical engineering, 15(6), 447-455.

Fossi, M., Egan, G., Haley, K., Johnson, E., Mack, T., Adams, T., ... & Wood, P. (2011). Symantec internet security threat report trends for 2010. Volume XVI.

Frankel, S. et al. (2005). Guide to IPsec VPNs: Recommendations of the National Institute of Standards and Technology, National Institute of Standards and Technology, USA, available at http://csrc.nist.gov/ publications/ nistpubs/ 800-77/sp800-77.pdf

Gepner, P., & Kowalik, M. F. (2006, September). Multi-core processors: New way to achieve high system performance. In International Symposium on Parallel Computing in Electrical Engineering (PARELEC'06) (pp. 9-13). IEEE.

Ghewari, P. B., Patil, J., & Chougule, A. (2010). Efficient hardware design and implementation of AES cryptosystem. International Journal of Engineering Science and Technology, 2(3), 213-219.

Gilbert, H., & Minier, M. (2000, April). A Collision Attack on 7 Rounds of Rijndael. In AES Candidate Conference (Vol. 230, p. 241).

Gilbert, H., & Peyrin, T. (2010, February). Super-Sbox cryptanalysis: improved attacks for AES-like permutations. In International Workshop on Fast Software Encryption (pp. 365-383). Springer, Berlin, Heidelberg.

Good, T., & Benaissa, M. (2005, August). AES on FPGA from the fastest to the smallest. In International workshop on cryptographic hardware and embedded systems (pp. 427-440). Springer, Berlin, Heidelberg.

Gueron, S. (2012). Intel® Advanced Encryption Standard (AES) New Instructions Set, available at https://software.intel.com/sites/default/files/article/165683/aes-wp-2012-09 22-v01.pdf

Gyanchandani, M., Rana, J. L., & Yadav, R. N. (2012). Taxonomy of anomaly based intrusion detection system: a review. International Journal of Scientific and Research Publications, 2(12), 1-13.

Hamalainen, P., Alho, T., Hannikainen, M., & Hamalainen, T. D. (2006, August). Design and implementation of low-area and low-power AES encryption hardware core. In 9th EUROMICRO conference on digital system design (DSD'06) (pp. 577-583). IEEE.

Hoban, A. (2010). Using Intel® AES New Instructions and PCLMULQDQ to Significantly Improve IPSec Performance on Linux, available at http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/aes-ipsec-performance-linux-paper.pdf

Intel (2012). Intel® Advanced Encryption Standard New Instructions (AES-NI), available at https://software. intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aesni/

Intel (2012). Intel® Core™ i7-4790K Processor (8M Cache, up to 4.40 GHz), available at http://ark.intel.com/products/80807/Intel-Core-i7-4790K-Processor-8M-Cache-up-to-4_40- GHz

Ishai, Y., Sahai, A., & Wagner, D. (2003, August). Private circuits: Securing hardware against probing attacks. In Annual International Cryptology Conference (pp. 463-481). Springer, Berlin, Heidelberg.

Jose, J. J. R., & Raj, E. G. D. P. (2012). A survey on the performance of parallelized symmetric cryptographic algorithms. International Journal of Research and Reviews in Computer Science (IJRRCS), 3(3).

Joye, M., Paillier, P., & Schoenmakers, B. (2005, August). On second-order differential power analysis. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 293-308). Springer, Berlin, Heidelberg.

Kaur, A., Bhardwaj, P., & Kumar, N. (2013). FPGA implementation of efficient hardware for the advanced encryption standard. International Journal of Innovative Technology and Exploring Engineering, 2(3), 186-189.

Kaur, S., & Vig, R. (2007, December). Efficient implementation of aes algorithm in fpga device. In International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) (Vol. 2, pp. 179-187). IEEE.

Kim, H. (2012). Security and vulnerability of SCADA systems over IP-based wireless sensor networks. International Journal of Distributed Sensor Networks, 8(11), 268478.

Kim, H., Hong, S., & Lim, J. (2011, September). A fast and provably secure higher-order masking of AES S-box. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 95-107). Springer, Berlin, Heidelberg.

Kim, J., Hong, S., & Preneel, B. (2007, March). Related-key rectangle attacks on reduced AES-192 and AES-256. In International Workshop on Fast Software Encryption (pp. 225-241). Springer, Berlin, Heidelberg.

Kim, J., Hong, S., Sung, J., Lee, S., Lim, J., & Sung, S. (2003, December). Impossible differential cryptanalysis for block cipher structures. In International Conference on Cryptology in India (pp. 82-96). Springer, Berlin, Heidelberg.

Koeune, F., & Quisquater, J. J. (1999). A Timing Attack against Rijndael. Université catholique de Louvain. Crypto Group, Technical report CG-1999/1.

Kömmerling, O., & Kuhn, M. G. (1999). Design Principles for Tamper-Resistant Smartcard Processors. Smartcard, 99, 9-20.

Krovetz, T., & Rogaway, P. (2011, February). The software performance of authenticated-encryption modes. In International Workshop on Fast Software Encryption (pp. 306-327). Springer, Berlin, Heidelberg.

Kumar, B. P., Ezhumalai, P., & Gomathi, S. S. (2010). Efficient implementation of a scalable encryption algorithm using FPGA. International Journal of Computer Applications, 3(10), 27-31.

Lamberger, M., Mendel, F., Rechberger, C., Rijmen, V., & Schläffer, M. (2009, December). Rebound distinguishers: Results on the full Whirlpool compression function. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 126-143). Springer, Berlin, Heidelberg.

Lee, J. G., Hwangbo, W., Kim, S., & Kyung, C. M. (2005, October). Top-down implementation of pipelined AES cipher and its verification with FPGA-based simulation accelerator. In 2005 6th International Conference on ASIC (Vol. 1, pp. 68-72). IEEE.

Lin, H., Slagell, A., Di Martino, C., Kalbarczyk, Z., & Iyer, R. K. (2013, January). Adapting bro into scada: building a specification-based intrusion detection system for the dnp3 protocol. In Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop (p. 5). ACM.

Lu, J., Dunkelman, O., Keller, N., & Kim, J. (2008, December). New impossible differential attacks on AES. In International Conference on Cryptology in India (pp. 279-293). Springer, Berlin, Heidelberg.

Lucks, S. (2004, February). Ciphers secure against related-key attacks. In International Workshop on Fast Software Encryption (pp. 359-370). Springer, Berlin, Heidelberg.

Mahmood, M. K., & Al-Naima, F. M. (2010). Developing a multi-layer strategy for securing control systems of oil refineries. Wireless Sensor Network, 2(07), 520.

Mala, H., Dakhilalian, M., Rijmen, V., & Modarres-Hashemi, M. (2010, December). Improved impossible differential cryptanalysis of 7-round AES-128. In International Conference on Cryptology in India (pp. 282-291). Springer, Berlin, Heidelberg.

Mangard, S., & Schramm, K. (2006, October). Pinpointing the side-channel leakage of masked AES hardware implementations. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 76-90). Springer, Berlin, Heidelberg.

Mangard, S., Pramstaller, N., & Oswald, E. (2005, August). Successfully attacking masked AES hardware implementations. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 157-171). Springer, Berlin, Heidelberg.

Matsui, M. (1993, May). Linear cryptanalysis method for DES cipher. In Workshop on the Theory and Application of of Cryptographic Techniques (pp. 386-397). Springer, Berlin, Heidelberg.

Medien, Z., Machhout, M., Bouallegue, B., Khriji, L., Baganne, A., & Tourki, R. (2010). Design and Hardware Implementation of QoSS-AES Processor for Multimedia applications. Trans. Data Privacy, 3(1), 43-64.

Mehra, P. (2012). A brief study and comparison of snort and bro open source network intrusion detection systems. International Journal of Advanced Research in Computer and Communication Engineering, 1(6), 383-386.

Mentens, N., Batina, L., Preneel, B., & Verbauwhede, I. (2005, February). A systematic evaluation of compact hardware implementations for the Rijndael S-box. In Cryptographers' Track at the RSA Conference (pp. 323-333). Springer, Berlin, Heidelberg.

Mentens, N., Batina, L., Preneel, B., & Verbauwhede, I. (2005, February). A systematic evaluation of compact hardware implementations for the Rijndael S-box. In Cryptographers' Track at the RSA Conference (pp. 323-333). Springer, Berlin, Heidelberg.

Mohan, G., & Rambabu, K. (2014). An Efficient FPGA Implementation of the Advanced Encryption Standard Algorithm. International Journal for Scientific Research & Development (IJSRD), 2(07), 413-417.

Nalini, C., Anandmohan, P. V., & Poornaiah, D. V. (2006, December). An FPGA based performance analysis of pipelining and unrolling of AES Algorithm. In 2006 International Conference on Advanced Computing and Communications (pp. 477-482). IEEE.

Nele Mentens IJCSNS International Journal of Computer Science and Network Security, VOL. 10 No.1, January 2010.

NIST (2013). Modes Development, National Institute of standards and Technology, USA, available at http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html

NIST Report on the development of the Advanced Encryption Standard (AES), National Institute of Standards and Technology, U.S.A., available at http://csrc.nist.gov/archive/ aes/round2/r2report.pdf.

NIST Special Publication 800-82 June 2011, Guide to Industrial Control Systems (ICS) Security. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

Njemanze, H. (2007). SCADA Security Protections Are On The Increase. Pipeline & Gas Journal, 234(2), 24-25.

Noo-Intara, P. (2004). Architectures for MixColumn Transform for the AES. Icep2004, ICEP.

O'Connor, L. (1993, May). On the distribution of characteristics in bijective mappings. In Workshop on the Theory and Application of Cryptographic Techniques (pp. 360-370). Springer, Berlin, Heidelberg.

Oswald, E., Mangard, S., Pramstaller, N., & Rijmen, V. (2005, February). A side-channel analysis resistant description of the AES S-box. In International workshop on fast software encryption (pp. 413-423). Springer, Berlin, Heidelberg.

Ozturk, M., & Aubin, P. (2011). SCADA security: challenges and solutions. Schneider Electric, 10.

Paul, R., Saha, S., Sau, S., & Chakrabarti, A. (2012). Design and implementation of real time AES-128 on real time operating system for multiple FPGA communication. arXiv preprint arXiv:1205.2153.

Ponemon Institute (2013). State of Endpoint Risk, Ponemon Institute, available at https://www.lumension.com/Lumension/media/graphics/Resources/2014-state-of- heendpoint

Rahman, T., Pan, S., & Zhang, Q. (2010, March). Design of a high throughput 128-bit AES (Rijndael Block Cipher). In Proceedings of International Multi Conference of Engineers and Computer Scientists (Vol. 2).

Rais, M. H., & Al Mijalli, M. H. (2012). Reconfigurable Implementation of S-Box Using Virtex-5, Virtex-6 and Virtex-7 Based Reduced Residue of Prime Numbers. World Applied Sciences Journal, 18(10), 1355-1358.

Rais, M. H., & Al-Mijalli, M. H. (2012). Field Programmable Gate Array Based Realization of S-Boxes. World Applied Sciences Journal, 18(10), 1343-1346.

Rais, M. H., & Qasim, S. M. (2009). Efficient hardware realization of advanced encryption standard algorithm using Virtex-5 FPGA. International Journal of Computer Science and Network Security, 9(9), 59-63.

Rais, M. H., & Qasim, S. M. (2010). Resource Efficient Implementation of S-Box Based on Reduced Residue of Prime Numbers using Virtex-5 FPGA. In Proceedings of the World Congress on Engineering (Vol. 2).

Regazzoni, F., Wang, Y., & Standaert, F. X. (2011, February). FPGA implementations of the AES masked against power analysis attacks. In Proceedings of COSADE (Vol. 2011, pp. 56-66).

Rivain, M., & Prouff, E. (2010, August). Provably secure higher-order masking of AES. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 413-427). Springer, Berlin, Heidelberg.

Rizk, M. R. M., & Morsy, M. (2007, December). Optimized area and optimized speed hardware implementations of AES on FPGA. In 2007 2nd International Design and Test Workshop (pp. 207-217). IEEE.

Rouvroy, G., Standaert, F. X., Quisquater, J. J., & Legat, J. D. (2004, April). Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well suited for small-embedded applications. In International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. (Vol. 2, pp. 583-587). IEEE.

Sasaki, Y. (2011, February). Meet-in-the-middle preimage attacks on AES hashing modes and an application to whirlpool. In International Workshop on Fast Software Encryption (pp. 378-396). Springer, Berlin, Heidelberg.

Satoh, A., Morioka, S., Takano, K., & Munetoh, S. (2001, December). A compact Rijndael hardware architecture with S-box optimization. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 239-254). Springer, Berlin, Heidelberg.

Schramm, K., & Paar, C. (2006, February). Higher order masking of the AES. In Cryptographers' track at the RSA conference (pp. 208-225). Springer, Berlin, Heidelberg.

Singh, A., Prasad, A., & Talwar, Y. (2016, October). SCADA security issues and FPGA implementation of AES—A review. In 2016 2nd International Conference on Next Generation Computing Technologies (NGCT) (pp. 899-904). IEEE.

Singh, G., & Mehra, R. (2011). FPGA based high speed and area efficient AES encryption for data security. International Journal of Research and Innovation in Computer Engineering, 1(2), 53-56.

Standaert, F. X., Peeters, E., & Quisquater, J. J. (2005, April). On the masking countermeasure and higher-order power analysis attacks. In International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II (Vol. 1, pp. 562-567). IEEE.

Standaert, F. X., Rouvroy, G., Quisquater, J. J., & Legat, J. D. (2003, September). Efficient implementation of Rijndael encryption in reconfigurable hardware: Improvements and design tradeoffs. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 334-350). Springer, Berlin, Heidelberg.

Stevens, K., & Mohamed, O. A. (2005, May). Single-chip FPGA implementation of a pipelined, memory-based AES Rijndael encryption design. In Canadian Conference on Electrical and Computer Engineering, 2005. (pp. 1296-1299). IEEE.

Stevens, K., & Mohamed, O. A. (2005, May). Single-chip FPGA implementation of a pipelined, memory-based AES Rijndael encryption design. In Canadian Conference on Electrical and Computer Engineering, 2005. (pp. 1296-1299). IEEE.

Talwar, Y., & Veni Madhavan, C. E. (2005). Rajpal Navin. On Partial Linearization of Byte Substitution Transformation of Rijndael–The AES. Journal of Computer Science, Science Publications, New York, USA, 2(1), 48-52.

Thulasimani, L., & Madheswaran, M. (2010). A single chip design and implementation of aes-128/192/256 encryption algorithms. International Journal of Engineering Science and Technology, 2(5), 1052-1059.

Thulasimani, L., & Madheswaran, M. (2010). A single chip design and implementation of aes-128/192/256 encryption algorithms. International Journal of Engineering Science and Technology, 2(5), 1052-1059.

Trichina, E., & Korkishko, L. (2004, August). Secure and efficient AES software implementation for smart cards. In International Workshop on Information Security Applications (pp. 425-439). Springer, Berlin, Heidelberg.

Uddin, M., & Rahman, A. A. (2010). Dynamic multi-layer signature based intrusion detection system using mobile agents. arXiv preprint arXiv:1010.5036.

Uskov, A. (2013, June). IPsec VPN-Based Security of Web-Based Rich Multimedia Systems. In IIMSS (pp. 31-40).

Uskov, A. (2014). The Efficiency of Encryption Algorithms in EAX Moder of Operation in IPSEC-based Virtual Private Networks for Streaming Rich Multimedia Data. IJCSA, 11(1), 18-36.

Uskov, A. V. (2012, June). Information security of IPsec-based mobile VPN: Authentication and encryption algorithms performance. In 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (pp. 1042-1048). IEEE.

Uskov, A., Byerly, A., & Heinemann, C. (2016). Advanced Encryption Standard Analysis with Multimedia Data on Intel® AES-NI Architecture. IJCSA, 13(2), 89-105.

Wagner, D. (1999, March). The boomerang attack. In International Workshop on Fast Software Encryption (pp. 156-170). Springer, Berlin, Heidelberg.

Wei, L., Rechberger, C., Guo, J., Wu, H., Wang, H., & Ling, S. (2011, July). Improved meet-in-the-middle cryptanalysis of KTANTAN (poster). In Australasian conference on information security and privacy (pp. 433-438). Springer, Berlin, Heidelberg.

Wiberg, K. C. (2006). Identifying supervisory control and data acquisition (SCADA) systems on a network via remote econnaissance. NAVAL POSTGRADUATE SCHOOL MONTEREY CA.

Wolkerstorfer, J., Oswald, E., & Lamberger, M. (2002, February). An ASIC implementation of the AES SBoxes. In Cryptographers' Track at the RSA Conference (pp. 67-78). Springer, Berlin, Heidelberg.

Wong, M. M., Wong, M. D., Nandi, A. K., & Hijazin, I. (2011). Construction of optimum composite field architecture for compact high-throughput aes s-boxes. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20(6), 1151-1155.

Zhang, W., Wu, W., & Feng, D. (2007, November). New results on impossible differential cryptanalysis of reduced AES. In International Conference on Information Security and Cryptology (pp. 239-250). Springer, Berlin, Heidelberg.

Zhang, X., & Parhi, K. K. (2004). High-speed VLSI architectures for the AES algorithm. IEEE transactions on very large scale integration (VLSI) systems, 12(9), 957-967.