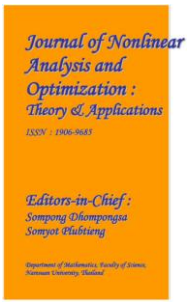
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.3 : 2024
ISSN : **1906-9685**



DESIGN OF PIPELINED ADVANCED ENCRYPTION STANDARD FOR DATA PROTECTION

Racharla Navatha, PG Scholar, Department of Electronics and Communication Engineering, CMR Engineering College (UGC – Autonomous), Kandlakoya, Medchal, Telangana, India
Dr. S. Poongodi, Professor, Department of Electronics and Communication Engineering, CMR Engineering College (UGC – Autonomous), Kandlakoya, Medchal, Telangana, India

Abstract
The data which is being communicated can be protected in different ways. For securing sensitive data in various industries Modified Advanced Encryption Standard (MAES) is used. To address issues, an automated key generation, power reduction, optimization of hardware resources techniques and mathematically less operation involved methods are proposed and analyzed in this work by employing modified substitution box algorithms for more security in present communication systems. The new S-box is developed by using the galois field using the affine transformation properties. By using this S-box, the pipelining mechanism was applied to the various stages of MAES. The pipelining mechanism effectively applied to the mix-columns, shift rows and key expansion modules respectively. As these blocks run parallel, it will reduce the time consumption for encryption process, the reduction in time causes to effectively reduces the power consumption for MAES system. The experiments conducted using Xilinx ISE environment, the simulation results show that the proposed method consumes lows hardware recourse utilization as well as power reduction compared to the conventional approaches.

Keywords:
Advanced Encryption Standard, Pipeline, Data Protection, Power Consumption.

1. Introduction

The interpretation of information to a mystery code is called encryption, which is used in the military and government to encourage mystery correspondence. The encryption is utilized in ensuring numerous sorts of non-military personnel frameworks, for example, Internet online business, mobile systems, programmed teller machine exchanges, duplicate security (particularly assurance against figuring out and Software robbery) and some more. The encrypted information must be deciphered on the off chance that one has the secret code or the key. The encryption algorithm is an arrangement of all around characterized ventures to change information from a meaningful configuration to an encoded organizes utilizing the key. This arrangement of very characterized advances is also called a cipher. An encryption algorithm gives privacy, verification, integrity and non - denial. Secrecy guarantees that the data is available for an individual's only approved arrangement. Validation is the demonstration of building up that the calculation is honest to goodness. Honesty when all is said in done means fulfillment however in encryption it is holding fast to some arrangement of standards. It depends on consistency with some numerical evidence. Non-denial in cryptology implies that it very well may be 2 confirmed that the sender and the beneficiary were, truth be told, the gatherings who professed to send or get the message, separately. Encryption calculations are comprehensively delegated symmetric or asymmetric calculations in view of the sort of keys utilized. Symmetric calculations have only one key. The sender who encrypts the information and the recipient unscrambles the information need a similar key. Asymmetric calculations have 2 keys, i.e. General Key and secret key. The information is encoded using the general key and is unscrambled by the receiver using the

secret key. The asymmetric algorithms are significantly increased mathematically when compared with the symmetric algorithms. There are still arguments about which kind of algorithm is safer but for speed and straight forwardness symmetric ciphers are constantly favored over asymmetric ciphers. Advance Encryption Standard, in view of the Rijndael algorithm is one such symmetric algorithm for encryption. These days transmission of data on web and security of data from strikes is one of the main problems. Keeping in mind the end goal to conquer these assaults encryption or decoding on information is important. In this manner AES is best appropriate calculation for encryption/unscrambling. It is important to fabricate calculation which counterparts the information rate. The cryptography causes in giving security towards information and this one likewise empowers towards storing and transmitting devoted data crosswise over troubled systems with the goal that the information can't be gotten to by unapproved individuals. The steadiness for incapable exchange of computerized information brings about a lot of various encryption measures. The system where the information is converted into a covert code is known as encryption. Intended for secret correspondence encryption strategy is actually utilized as a part of military administration zones and furthermore this method is utilized via government on behalf of secret information transmission. In numerous types of resident frameworks, for example, mobile systems, online business, automatic teller machine exchanges, copy security namely programming theft encryption system is actually utilized.

In communication network safety is very essential in order to transmit and receive audio, video and picture. This work is to establish a new approach of the Advance Encryption standard algorithm with a fuzzy concept principle used to control S box and also its FPGA implementation for actual performance factors such as speed and area for security concern. The prime objective of this research is to design and development of FPGA based Novel approach of AES algorithm cryptography systems for high security and low power consumption to transfer data from one router to another. Use of suitable and efficient algorithms/techniques will definitely improve in terms of speed, power, delay and security. The proposed system certainly will upgrade the overall performance of the present cryptography system. The proposed approach has efficient characteristics for high security and low power consumption includes novel techniques such as modified AES algorithm.

Rest of the paper as follows; section 2 gives the detailed analysis various literature methods and its drawback. Section 3 gives the detailed operation of proposed pipelining used for the MAES operation. Section 4 gives the detailed analysis of proposed MAES with detailed analysis of each block. Section 5 gives the detailed analysis on results with respect to both simulation and synthesis outcome and comparative analysis also performed with various convention approaches. Section 6 deals about conclusion and future works of proposed methods.

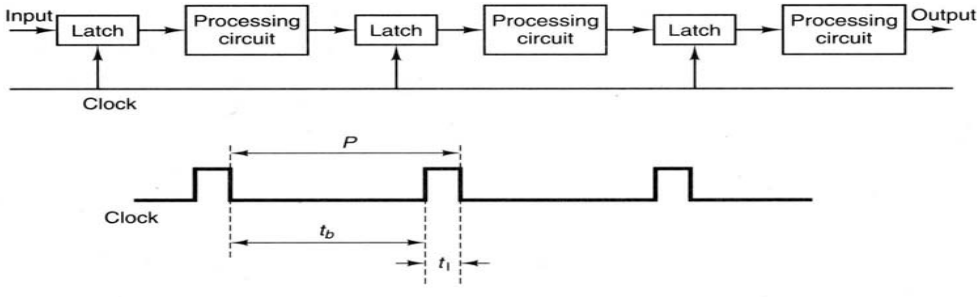
2. Literature survey

There are many research articles published in the last five years focusing surveys on lightweight cryptographic algorithms. Most of the research paper targeted at cryptographic properties and Implementation properties include Block Size, Key Size, Cryptographic Structure, Number of Rounds, and cryptanalysis attacks. Hardware implementation properties include Technology, Footprint area utilization, Throughput and Power Consumption. In [1-2] authors have experimented various lightweight cryptographic algorithm for IoT. Performance analysis includes data size, power consumption, memory space, and computation power were the key area focus. In [3-4] authors focus on benchmarking of lightweight cryptographic algorithms on the legion of embedded systems. The authors have evaluated execution of several lightweight block cipher on three platforms 8-bit Atmel AVR microcontroller, 16-bit ultra-low power MSP430 microcontroller, and 4-byte processor CortexM. The authors have computed execution time, footprint area, RAM consumption, and computing code size of several algorithms. The implementation of each block cipher was done on ANSI to support portability for IoT and assembly implementation for AES and PRESENT. In [5-6] authors conveyed outline and advancement of improved circuits of various lightweight block ciphers. The test conclusion drives blend stage and reproduction process on 14nm FinFET CMOS technology. The authors specifically focus on using the lightweight cryptographic algorithm for providing security to Automotive, Internet of Things, and Wireless Sensors Devices. The authors aim at optimizing architectures of 4 lightweight block cipher: PRINCE, SIMON, SPECK and PRESENT with footprint

area utilization is 1278, 1358, 1244, 934 gates equivalent respectively. It also provides high security in embedded system environment. In [7-8] authors strongly focus on cryptanalysis of lightweight block ciphers. The author carried successful analyze the transmission of information between embedded systems and monitor the physical leakage during cryptographic operation in IoT device/nodes. The side channel attack allows determining the intermediate value using a secret key. Particularly, the author focuses on AES and PRESENT lightweight block cipher. The study also shows the difference in side-channel flexibility using 4-bit and 1-byte S-boxes. Evaluation shows S-box of KLEIN is hardest. The authors proposed to make use of 4-bit S-Boxes for implementing lightweight block cipher which ultimately minimizes area and cost. In [9-10] authors critically analyzes IEEE standard P1735 and uncovered major issues which can be exploited to decipher information from electronic encrypted system-on-chips (SOC). Such attacks enable an assailant to recoup all the plaintext data. The critical vulnerability lies in the cipher-block-chaining mode in AES algorithm by which the confidentiality can reveal entire plaintext information. This exploitation leverages a profitable tool for System on a Chip (SoC) tool. The authors also proof various attacks which include the cryptographic mechanism for IP licensing. They also verify that using IEEE standard P1735 would consider themselves at exposure to attackers. In [12-15] authors present a comprehensive evaluation of lightweight cryptographic primitives in terms of cost of algorithms in gate equivalent (GE), Operating speed in metric of clock cycle per unit block, throughput performance of algorithms, and balanced efficiency of various lightweight block ciphers. The authors also conclude that SIMON, SPECK and PICCOLO lightweight block cipher suits best when it comes to hardware implementation because of its low cast implementation.

3. Pipelined technique for MAES algorithm

Over the past few years, many researchers around the world have done tremendous research in the field of the lightweight block cipher. However, most of the analysts tried the hands in simulating their designs into working projects using very high-speed integrated environment like VHDL, ASIC design, Verilog Programming and other programming tools. Few of the researchers focused on hardware implementation of their design over Field Programmable Gate Arrays (FPGA) due to low requirement of hardware resources. It is important for researchers to implement their design over hardware platform to test real time data and information processing over power constrained devices and embedded systems. Focusing on our compact implementation of MAES encryption algorithm architectures with block size of 128-bit that were designed by using modified S-BOX for performance comparison. Furthermore, the internal structure of compact S-boxes is also designed by using pipelined architectures. The design of compact lightweight MAES algorithm has deployed using dynamic configuration, Internal data replication, inner-loop pipelining, and loop unrolling techniques. A method of implementing parallel-pipeline architecture to our MAES algorithm with 128-bit block size, since widely held research studies have demonstrated the prevalence of several lightweight block cipher over conventional block cipher schemes including the MAES encryption algorithm for all intents and purposes utilized by international standard in wireless communications. The parallel-pipeline technique used in our MAES architecture to expand the throughput and decrease the power utilization as shown in figure 1. The encompassment of the pipelined technique is implemented in our MAES encryption algorithm design to control internal signals of data processing in powered S-boxes and Feistel function. This was done to minimizing critical path delay, latency and power utilization.



P: Clock period.
t_b: Maximum time for a stage to perform a function.
t_l: Time for latch to accept input data.

Figure 1: Proposed Pipeline mechanism of MAES algorithm

The input data is applied to the latch unit and all the latches are controlled by the same clock. The latch will stores the data during the positive edge triggering time t_1. The preprocessing units will analyze the data during the time period t_b, it is the maximum time period of the clock. A total of three stages of latch and preprocessing operations will be done to store the data generated from MAES operation. The primary purpose of using Pipeline mechanism is to improve throughput and performance. We have also developed a file encryption tool in sequential and parallel methods. Standard CSL is a sequential algorithm using different block cipher modes of operation. Any sequential algorithm can execute much faster by using parallelization with the appearance of parallel processors for computing. Intel and AMD processors allow multi-processing by use of parallel programming application programmable interface such as OpenMP. Practically, it is now feasible to make use of any cryptographic algorithm: that runs in parallel by specifying its parallel task to run on different threads of CPU to minimize time complexity. Pipeline mechanism divides the plaintext block into fixed size block and perform encryption on a series of block concurrently for multiple number of rounds on the multi-core machine to generate ciphertext as an output of MAES. Another important aspect of using pipelining approach is to use the pipelined registers available on FPGA hardware implementation, with the help of 3-layer pipelining in composite field arithmetic powered S-box is proposed to break the logic and to achieve high clock frequency. Pipelining is a vital idea to guarantee the planning is precise for different correspondences and control applications by minimizing critical paths in the system to achieve high performance without overheads of latency and power utilization. The transportations interfaces that are used to transmit and receive the MAES encrypted data are fully pipelined. The MAES algorithm works parallel with compact S-box for key expansion process in accordance with internal functions used in the Feistel network.

3. Proposed methodology

AES is a symmetric algorithm as both encryption and decryption processes use the same key. It also separates the plain text into blocks for processing. So, it is called a symmetric block cipher algorithm. This algorithm uses fixed input size blocks of 128 bits called data blocks. The strength of the AES algorithm is that it encrypts the information of the input data block and hides the correlation between the input, the output and the secret key. The MAES algorithm has two mathematical functions: a function for encryption (AESenc) and a function for decryption (AESdec). Modified Advanced Encryption Standard Algorithm (MAES) is a symmetric key cryptographic performing a combination of four types of sub-operations in each round for encryption and decryption namely, AddRoundKey, SubBytes, ShiftRows, and MixColumns. Encryption process consists of three different processes performed in an initial round, a fixed number of iterations of a main round, and ends with a final round. All three types of rounds employ the same set of Sub-operations performed in varied orders as shown below figure 2.The repetition of the main round depends on the variant of MAES employed. MAES-128 repeats the main round for nine times, MAES-192 for eleven times and MAES-256 for thirteen times.

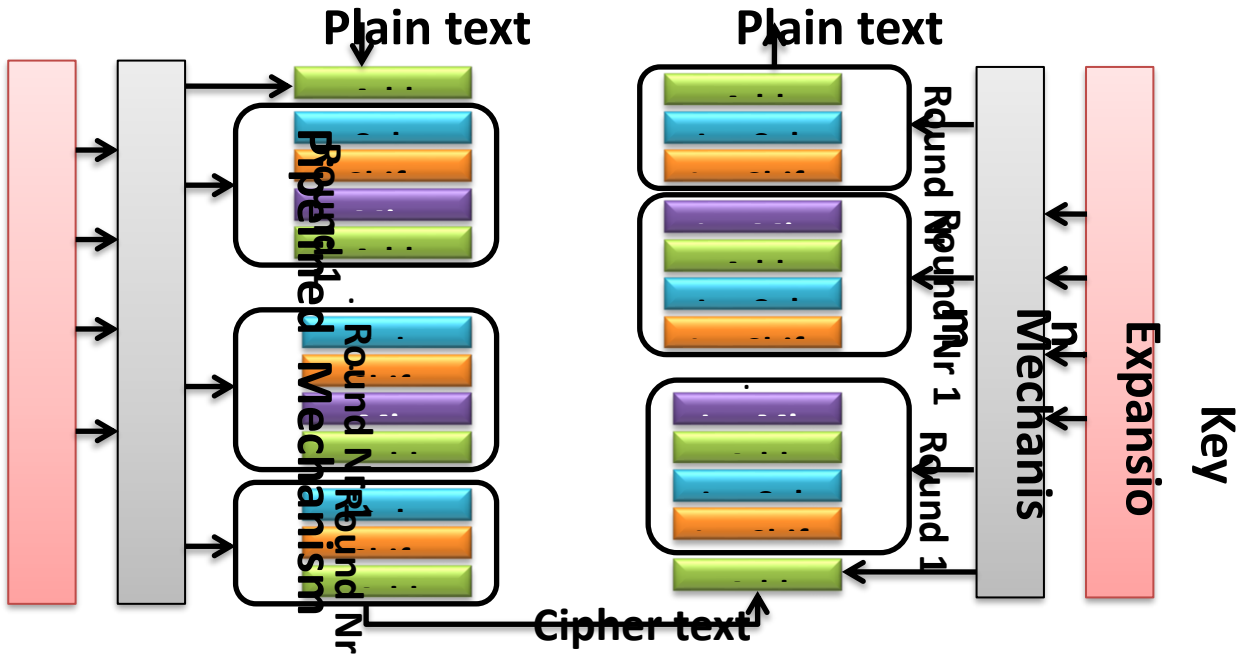


Figure 2: Proposed MAES Encryption and Decryption Process

ADD ROUND KEY

The size of MAES key may be 16 bytes, 24 bytes or 32 bytes and this determines the type of AES employed. The key is operated upon directly only in the Add Round Key operation in the entire MAES encryption process. This operation performs XOR (Exclusive-OR) operation on the input to the round and the key provided for that particular round.

SHIFTROWS

The internal state of the MAES cipher is a 16 bytes block and it is arranged in the form of a matrix having 4 rows and 4 columns. The 16 bytes are arranged in the matrix left to right from top to bottom. Each of the four rows in this matrix representing the internal state of the AES cipher is shifted left by a fixed amount in the Shift Rows operation. The row number ranges from 0 to 3 and the shift amounts are determined by the row number. 0th row is not shifted, 1st row is left shifted once, 2nd row is left shifted twice, and 3rd row is left shifted thrice with ai, j represents the byte in the ith row and jth column, as described in figure 3.

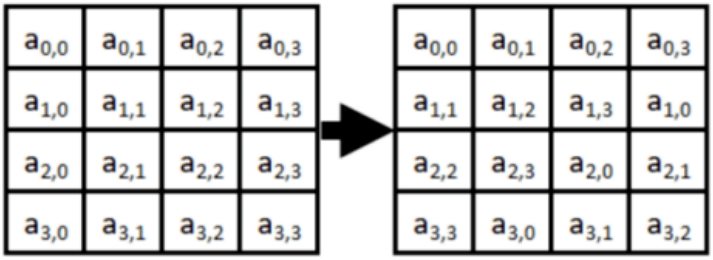


Figure 3: Matrix Column Shifting Process

MIXCOLUMNS

The Mix Columns operation mixes the input around like Shift Rows operation. This operation also manipulates the internal state matrix but unlike Shift Rows operation that manipulates the rows, the Mix Columns operation manipulates the columns as the name specifies. The Mix Columns operation is represented in a visual manner as shown in figure 4 below.

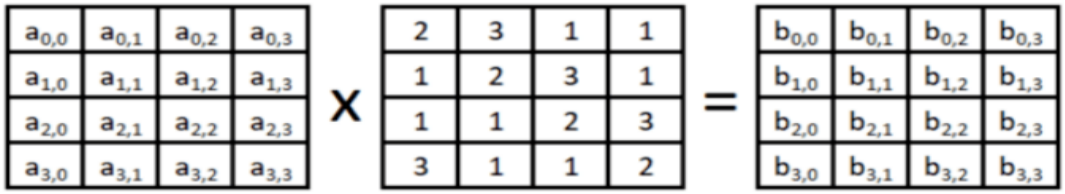


Figure 4: Matrix Mix columns Operations

Each column of the 16 bytes block arranged as a matrix is pre-multiplied by the specified matrix to yield the modified column for the output. This operation is repeated for all columns of the matrix and finally all the outputs are arranged once again as a matrix by combining all columns. This operation ensures that any change in any one of the input byte is reflected in all four bytes of the output. This operation utilizes matrix multiplication based on Galois Field (2^8) unlike normal matrix multiplication. The multiplication of each column with the specified matrix is performed independently and thus can be performed simultaneously in parallel to safe time.

MAES KEY EXPANSION

A unique key for each round is generated from the initial key with the MAES Key Schedule process. The original key is used as such only for the AddRoundKey operation of the initial round of the MAES encryption process. For the remaining rounds, distinct keys are generated by the MAES Key schedule process and provided as input to the AddRoundKey operation of each round. If MAES-128 version is used ten different keys are generated one each for the nine main rounds and one final round. If MAES-192 version is used twelve different keys are generated one each for the eleven main rounds and one final round. If MAES-256 version is used fourteen different keys are generated one each for the thirteen main rounds and one final round, with each cell represent a byte of the key. Top row shows the bytes of the key used in the previous round and bottom row shows key produced for the current round. The figure assumes that 128-bit key is utilized for this encryption. The original key is provided as input for the first round. The row is split into words (1 word = 32 bits = 4 bytes). Each word of the input key is XOR-ed with another word of the same key and produces the new word for the output key of the present round. The entire key is thus split into four words (4 x 32 = 128). The words are numbered 0, 1, 2 and 3. For generating each word of the output key, the previous word of the input key, which is the key produced as output in the previous round, is used as one value of the XOR operation. For the words 1, 2 and 3, one operand for the XOR operation is the key words 0, 1, and 2 of the preceding round respectively. For the word 0 since there is no previous word, the last word is used after being processed by the galios function as shown in the figure 4. The operations used in the S-Box of the key schedule process are the same as that used for encrypting the text. The permutation operation in the S-Box involves left shift of each word by one position. The last operation in the g ( ) function is XOR which involves the leftmost byte and a distinct constant for each round. The current key produced by this scheduling function in each round is used in the AddRoundKey operation of that round. For each round, the same process is repeated to generate the current round key from the previous round key.

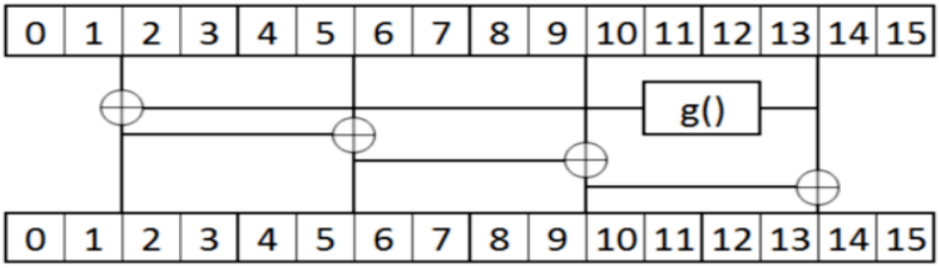


Figure 5: Process of key expansion

MAES S-Box generation:

The input is split into bytes and then taken through an S-Box (Substitution Box) in the SubBytes operation. AES sub-Bytes operation employs the same S-Box for all bytes unlike DES which employs different S-Boxes. The substitution box of MAES employs inverse multiplications in Galois Field (2^8). The new byte to be substituted for an input byte is decided with this 16 x 16 table. The input byte is first broken into two equal parts, each having 4-bits. The first part is used to select one row among the sixteen different rows in the table and the second part is used to select one column among the sixteen different columns in the table. S-Box is a function that maps m input bits into n output bits where n and m may or may not be equal. S-Box Si: {0, 1} m → {0, 1} n Substitution boxes are basic components of the symmetric key systems. This is a 2m x 2n table in which every column represents output and every row represent input difference. Generally a fixed table is used as in case of DES but some newer design are using variable tables that are dynamically generated using the key. A carefully designed S-Box can thwart the linear and differential cryptanalysis attacks. Let X = (I_0, I_1, ... I_{m-1})

be a input vector of S-Box and let F = (O_0, O_1, ..., O_{n-1}) be the output vector. Then the S-Box output can be derived by these equations:

$$F = M(\delta^{-1}(\delta X)^{-1}) + V$$

Here, X^{-1} denotes the Affine Transformation, M denotes the function of S-box and V denotes the constant data vector. Where M is mapping 2 m→ 2 n , and 2 is a binary field. Affine Transformation has defined some basic criteria for selection and generation of good SBox. Multiplicative Inverse Table said that some good pseudorandom number generator should be used for generating S-Box contents and these contents should be thoroughly tested against different design criteria for acceptance or rejection. Mathematical principles should to be used for S-Box generation so that it can provide good diffusion properties and can be secure against linear and differential cryptanalysis. It explains the shift registers and other basic building blocks of synchronous stream cipher design in detail. We have successfully shortened the critical path required in our compact S-box. The implementation of S-box is based on Composite Field Arithmetic technology implanted to map the MI (Multiplicative Inverse) in our S-box from finite field Galois Field GF(2^8) to composite field GF((2^4) 2). It was proved by S. Morioka et al. that the S-Box implemented using the composite field arithmetic technology requires fewer Lookup Tables (LUT) and has smallest hardware footprint area cost. The implementation of SBox utilized as one proceeds with way would increase logic delay, due to this it will reduce the most elevated clock frequency.

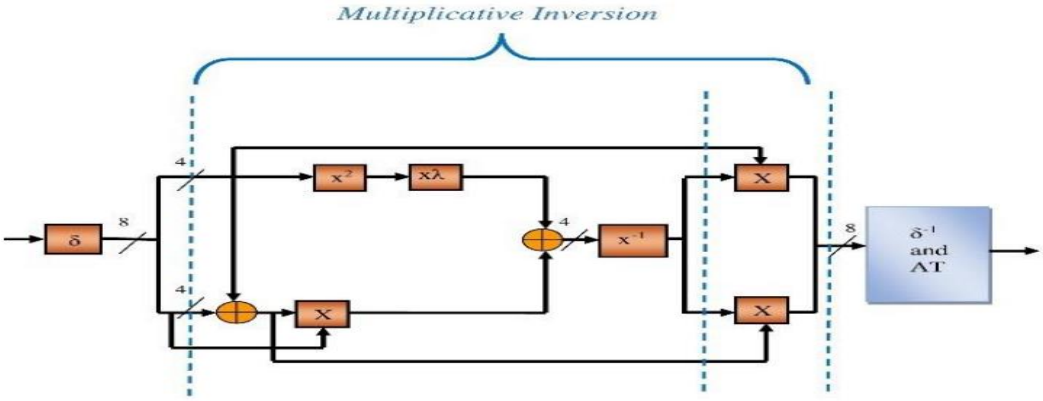


Figure 6: Internal structure of compact S-Box

Here, δ Isomorphic mapping, X^2 Squarer in $GF(2^4)$, X^{-1} Multiplicative Inverse in $GF(2^4)$,AT- Affine Transformation in $GF(2^4)$, $\oplus$ Addition $GF(2^4)$, δ^{-1} Inverse Isomorphic mapping to $GF(2^4)$, X Multiplication Operation in $GF(2^4)$ and $X\lambda$ Multiplication with Constant in $GF(2^4)$.

MAES 128-bit Encryption

The MAES encryption (AESenc) is series of matrix calculations which are repeated ten times, each of these repetitions is called rounds. For each round is the state matrix "mixed" with the key, the result is a new state matrix and a new key which is used as input to the next round. The MAES Encryption is a rather complicated series of matrix operations. The order of the different operation is illustrated in Figure 7.

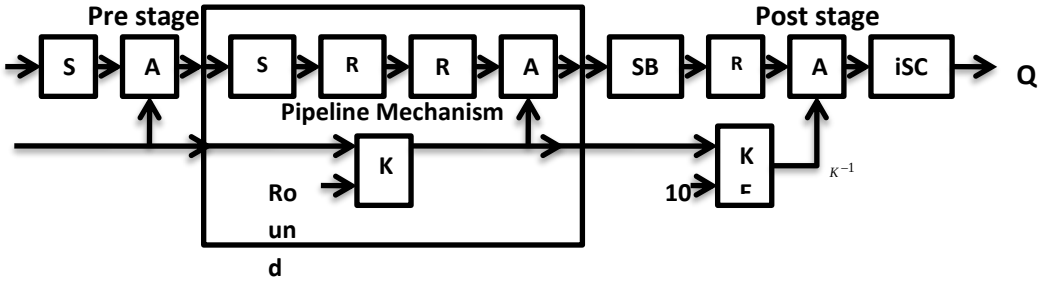


Figure 7: Structural Diagram of MAES 128 Encryption

The pre stage consists of a state matrix conversion, to convert the 128 bit input vector to a 4x4 byte matrix followed by an Add key operation. The input key is used for add key operation as well as transferred to the round 0. Round 0 to 9 are cascaded, meaning that the same parallel of operation is performed ten times. The output produced by each round in MAES is provided as the input to the succeeding round. Each round constitutes of a substitution operation with the help of an S-Box, a shift row, mix column and add key operations. The key for the given round number is calculated with the key expander, the new key is used as input to the following round. The output from last round (round 9) is transferred to the post stage. The post stage is the final round for the encryption, it consists of a substitution box, shift row and add key operation. The key for the last add key operation is calculated as well with a key expand operation. The last key calculated is in fact K 21, however the key is not output under normal circumstances, since the key has the same level of confidentially as the input key. By having the K21, K can be found simply by performing 11 rounds of inverse key expanding

MAES 128-bit Decryption

The MAES decryption (AESdec) is the same series of matrix operations as for MAES encryption; however the operations are mathematically inverse and are performed in opposite order, as shown in Figure 8.

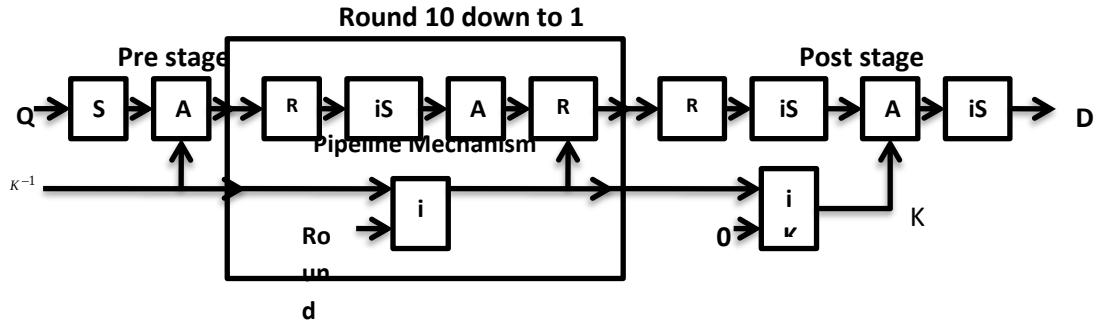


Figure 8: Structural Diagram of MAES 128 Decryption

The pre stage consists of a state matrix conversion, to convert the 128-bit input vector to a 4x4 byte matrix followed by an Add key operation. The Add Key operation can be reversed by performing it. Therefore, it is the same Add Key operation used in encryption. The inverse key is used for adding key operation as well as transferred to round 10. Round 1021 is cascaded in the same way as for encryption, the difference is that we start with round 10 and go down to round 1 2 the post stage will process round 0, which is the last round for decryption. The output produced by a round is provided as input to the successive round. Each round consists of an inverse shift row, inverse substitution box, add key and inverse mix column operation. The key for the given round is calculated with the key expander, the new key is used as input to the following round. The output from last round (round 1) is transferred to the post stage. The post stage consists of an inverse shift row, inverse substitution box and a final add key operation. The key for the last add key operation is calculated with a key expand operation. The last key calculated is in fact K1.

5. RESULTS

All the proposed designs have been programmed and designed using Xilinx ISE software this software tool provides the two categories of outputs named as simulation and synthesis. The simulation results give the detailed analysis of proposed design with respect to inputs, output byte level combinations. Through simulation analysis of accuracy of the addition, multiplication process estimated easily by applying the different combination inputs and by monitoring various outputs. Through the synthesis results the utilization of area with respect to the programmable logic blocks (PLBs), look up tables (LUT) will be achieved. And also, time summary with respect to various path delays will be obtained and power summary generated using the static and dynamic power consumed.

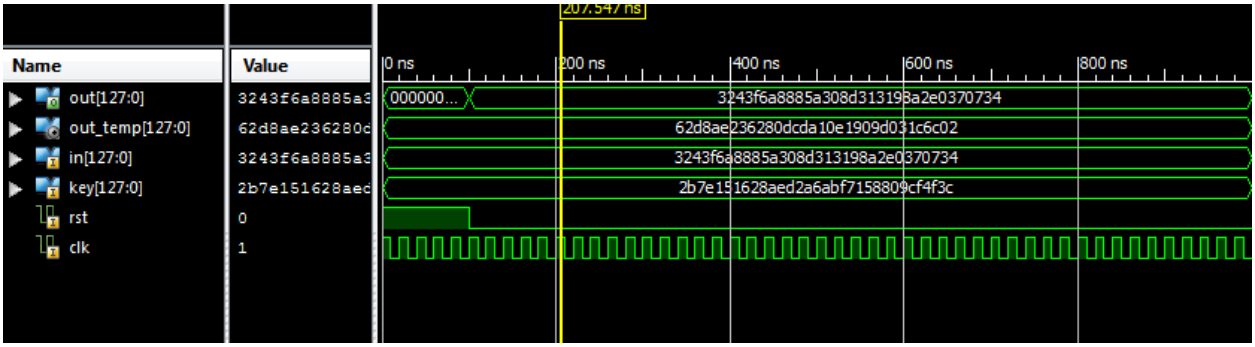


Figure 9: Simulation output

The above result represents the simulation waveform by using the Xilinx ISE software. Here CLK, RST are the control inputs. In and Key are the 128-bit key inputs, after performing the encryption operation out_temp generated. And after successful completion of decryption operation out will be generated, which is same as the applied input data. Here for encryption and decryption same key will be used respectively.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	8472	63400	13%
Number of fully used LUT-FF pairs	0	8472	0%
Number of bonded IOBs	386	210	183%
Number of BUFG/BUFGCTRLs	1	32	3%

Figure 10: Design summary

The above result represents the synthesis implementation by using the Xilinx ISE software. From the above table, it is observed that only 8472 look up tables are used out of available 63400. It indicates less area was used for the proposed design.

```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: 8.854ns (Maximum Frequency: 112.943MHz)
   Minimum input arrival time before clock: 52.495ns
   Maximum output required time after clock: 13.040ns
   Maximum combinational path delay: 17.849ns
```

Figure 11: Time summary

The above result represents the time consumed such as path delays by using the Xilinx ISE software. The consumed path delay is 41.345ns.

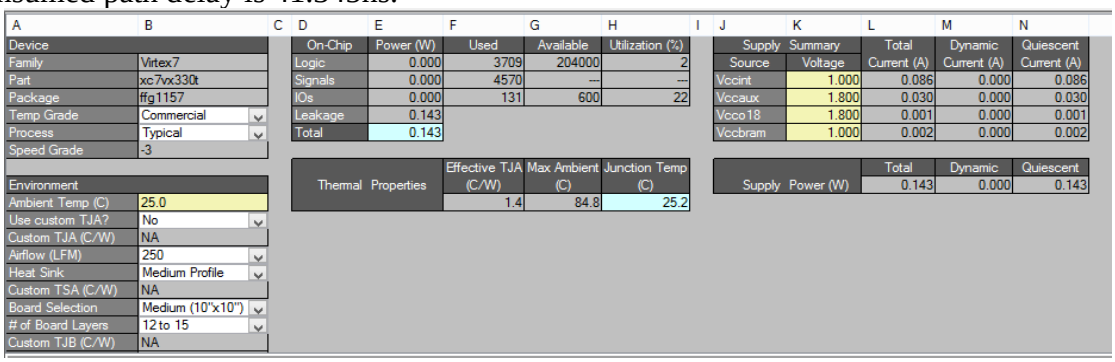


Figure 12: Power summary

The above result represents the power consumed by using the Xilinx ISE software. The consumed power is 0.143uw. From the table 1 it is observed that the proposed method consumes very less area compared to the conventional approaches such as MAES [1],FAES [4] and QAES [3]respectively.

Table 1: Comparison table

Parameter	MAES [1]	FAES [4]	QAES [3]	**PROPOSED METHOD**
Time delay	63.13 ns	66.10 ns	59.110 ns	**17.345 ns**

Power utilized	2.364 uw	2.12uw	1.293uw	**0.143 uw**
Look up tables	28108	26037	21029	**8472**
Area utilized	78%	63%	45%	**13%**

6. CONCLUSION

As the side channel attack is a passive and noninvasive attack, it is very difficult to protect a system from such an attack. The framework of the proposed architecture is also compared with the existing cryptography algorithms which are developed for cloud storage and transmission of data. The proposed MAES technique was developed using pipelining mechanism and different experimental analyses are conducted and tested to examine the performance and its reliability using the FPGA environment. The work presented in this work can be tested on real life and extended towards the directions to implement the same proposed technique to the virtual migration sector in cloud computing. Since the proposed secure double encryption is applied only for side channel attacks, it can be further applied to various kinds of threat attacks happening over the cloud.

References

[1]Chowdhury, Arnab Rahman, et al. "MAES: modified advanced encryption standard for resource constraint environments." *2018 IEEE Sensors Applications Symposium (SAS)*. IEEE, 2018.

[2]Khan, Majid, and Noor Munir. "A novel image encryption technique based on generalized advanced encryption standard based on field of any characteristic." *Wireless Personal Communications* 109.2 (2019): 849-867.

[3]Langenberg, Brandon, Hai Pham, and Rainer Steinwandt. "Reducing the Cost of Implementing the Advanced Encryption Standard as a Quantum Circuit." *IEEE Transactions on Quantum Engineering* 1 (2020): 1-12.

[4]Gueron, Shay, et al. "Flexible architecture and instruction for advanced encryption standard (AES)." U.S. Patent No. 10,554,386. 4 Feb. 2020.

[5]Shreedhar, A., et al. "Low gate-count ultra-small area nano advanced encryption standard (AES) design." *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019.

[6]Rupanagudi, Sudhir Rao, et al. "A further optimized mix column architecture design for the advanced encryption standard." *2019 11th International Conference on Knowledge and Smart Technology (KST)*. IEEE, 2019.

[7]Leech, David P., Stacey Ferris, and John T. Scott. "The economic impacts of the advanced encryption standard, 1996–2017." *Annals of Science and Technology Policy* 3.2 (2019): 142-257.

[8]Seghier, Athmane, and Jianxin Li. "Advanced encryption standard based on key dependent S-Box cube." *IET Information Security* 13.6 (2019): 552-558.

[9]Murtaza, Abid, et al. "Parallelized Key Expansion Algorithm for Advanced Encryption Standard." *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2019.

[10] Gill, Ishpal Singh, and Dharm Singh Jat. "Advanced encryption standard with Randomized round keys for communication security in IoT networks." *The IoT and the Next Revolutions Automating the World*. IGI Global, 2019. 280-288.

[11] Gamido, Heidilyn V. "Implementation of a bit permutation-based advanced encryption standard for securing text and image files." *Indonesian Journal of Electrical Engineering and Computer Science* 19.3 (2020): 1596-1601.

[12] Novianto, Dian, and Yohanes Setiawan. "Aplikasi Pengamanan Informasi Menggunakan Metode Least Significant Bit (Lsb) dan Algoritma Kriptografi Advanced Encryption Standard (AES)." *Jurnal Ilmiah Informatika Global* 9.2 (2019).

[13] Rahma, Abdul Monem S., and Atheer M. Abbas. "A modified Matrices Approach in Advanced Encryption Standard Algorithm." *Engineering and Technology Journal* 37.3B (2019): 86-91.

[14] Shakya, Siddhartha, et al. "Novel secure surgical telepresence using enhanced advanced encryption standard: during, pre and post surgery." *Multimedia Tools and Applications* (2020): 1-26.

[15] Teng, Lin, et al. "A Modified Advanced Encryption Standard for Data Security." *IJ Network Security* 22.1 (2020): 112-117.