Secured AES 256 Using Verilog

Amal K. B [1], Rachana M. K [2]

[1] *M.Tech Scholar, Electronics & Communication Engineering, IES College of Engineering, Kerala, India*
[2] *Assistant Professor, Electronics & Communication Engineering, , IES College of Engineering, Kerala, India*
Email_id: rachanamk@iesce.info

Abstract

Although many algorithms were originally developed to encrypt and decode data, previous methods are not effective in protecting large amounts of sensitive information. As a result, AES was developed as a new standard for data encryption and decryption. AES was initially used primarily to encrypt highly sensitive data, but later became the industry standard for data protection for many Internet applications. Its main purpose is to protect sensitive data, and it is sometimes used to improve data security in backend network systems. Verilog offers much faster runtime and propagation delay for both data encoding and decoding than other HDL languages, which is the main argument for switching from standard VHDL to it. Before AES, DES was used as the encryption standard. The main drawback of DES is that the fixed key size of 56 bits. To make it more secured AES with key size 256bit size used.

Keywords: *Advanced Encryption Standard, Low Power, Secured, Key Size, S Box.*

DOI: https://doi.org/10.5281/zenodo.15010319

1. Introduction

Cryptography is the art of converting plain and understandable English into unintelligible writing and vice versa. Cryptography is the study of encryption or hidden writing. The three types of encryption methods are symmetric key encryption, hash functions, and public key encryption. One of the inputs to the algorithm is its secret key. The key is a string or an integer. If an algorithm uses the same key for both encryption and decryption, it is called symmetric; if not, it is called asymmetric. In an asymmetric algorithm, two keys are used: a public key for encryption and a private key for decryption. Messages encrypted with the public key can only be decrypted with the private key.

Symmetric key technologies such as Advanced Encryption Standard (AES) and Data Encryption Standard use the same key for both encryption and decryption. It is significantly faster, easier to implement and requires fewer computing resources. Mathematical strategies and methods known as cryptographic algorithms help protect data. Bit-by-bit or byte-by-byte data is changed using stream encryption. On the other hand, block ciphers convert data into large chunks (64 or 128 bits) at a time. Block ciphers are considered one of the best methods for protecting data in modern symmetric cryptography. Modern block ciphers include Advanced Encryption Standard (AES), Blowfish, and Data Encryption Standard (DES). Compared to stream ciphers, block ciphers are easier to construct precisely and have a wider range of uses.

The National Institute of Standards and Technology (NIST) is the authority that produces cryptographic recommendations and guidelines. NIST launched an open competition that led to the development of the Advanced Encryption Standard (AES). Demon and Rags later recognized AES as the winner. AES is used by many programs

today. It is an encryption algorithm that uses a secret key of the same size to convert plaintext to ciphertext. In December 2001, the National Institute of Standards and Technology (NIST) published the Advanced Encryption Standard (AES), a symmetric key block. Available in three different lengths. There are ten encryption/decryption rounds for 128-bit keys, twelve rounds for 192-bit keys, and fourteen rounds for 256-bit keys.

Parameters	DES	AES
Key Size	56	128
Block Size	64	128
Rounds	16	10

Table 1: comparison of DES and AES

DES is the name of an obsolete encryption and decryption method. However, its use is often discouraged due to its relatively weak protection compared to more advanced encryption methods. Additionally, the National Institute of Standards and Technology has banned the use of DES by government agencies since 2005.Two popular symmetric encryption methods are Advanced Encryption Standard (AES) and Data Encryption Standard (DES). Although there are some similarities between the two, there are also many notable differences shown in table 1. Key size is one of the main differences between AES and DES. AES can handle 128-, 192 or 256-bit keys, while DES can only handle 56-bit keys. Because AES has a larger key size than DES, it is therefore more secure because it is harder to crack [1]. The National Institute of Standards and Technology (NIST) notes that most applications actually prefer AES over DES.

The block strengths of AES and DES also differ. AES uses a 128-bit block size, while DES uses a 64bit block size. This suggests that AES can sometimes be faster than DES because it can encrypt more data in one operation. The encryption methods used by DES and AES are different. DES divides the input into two parts using a Feistel network. Alternatively, AES uses a replacement-permutation network that continuously reorders and replaces incoming data. AES is generally considered more secure than DES. DES has been shown to be vulnerable to brute force attacks, where an attacker tries every key until they find one that works. AES is much more resistant to such attacks due to its larger key size and more advanced encryption method.

Although DES was once widely used and considered secure, AES has been overtaken by DES as a symmetric encryption technology in many situations. This is due to the larger block size, larger key size and better security of AES. However, DES is only used in a few applications. AES provides a high level of security by combining propagation, substitution, and permutation functions to convert plaintext data into ciphertext that cannot be read without the correct key. Key security-enhancing features of AES are key management, encryption, decryption, and generation. These processes ensure that plaintext data cannot be read without the correct key, enabling the advanced level of AES security. The length of the secret and the number of encryption cycles used determine how secure AES is. AES supports key lengths up to 256, 192 and 128 bits. The length of the key can affect the number of revolutions. The longer the key, the higher the level of AES protection and the more rounds are used.

2. Literature Review and Objective

The security concerns raised in research have been utterly disregarded, and only the execution of the most popular secret key algorithms include contrasted across several platforms, employing input files with different formats and amounts of data. When evaluating how well algorithms perform, the time needed to configure the key or keys has

been neglected. The trials have shown that, of the methods selected for implementation, the Blowfish algorithm performs the best. But by considering security as well and also other aspects AES is much better [1].

Verilog is faster than VHDL in terms of clock cycles and time. Fewer clock cycles reduce power usage. AES is harder to hack or damage since it performs more operations every round than DES. DES has a smaller key size (56 bits) than AES. AES can be implemented for verification using the Verilog system; it has greater benefits than Verilog. High-speed real-time applications benefit greatly from hardware implementation. It increases adaptability [4].

Numerous studies demonstrate AES's superior usability over other available algorithms. Empirical research and theoretical analysis have shown that the AES technique provides low data transfer across open channels along with rapid speed [2]. It's important to realize that there is no maximum length for the secret key. The system gets more secure and allows unauthorized users more privacy when the number of rounds is increased. The additional number of rounds will take longer to process, increasing the difficulty for hackers to infiltrate the system. FPGA was used in the hardware implementation of AES to obtain fast throughput and reduced critical delay. Additionally, it reduces hardware complexity by utilizing the FPGA's pipelining implementation properties in AES. Similar to how using a different architecture for the S-box enhanced security, combining encryption and decryption processes at specific points might also further minimize hardware complexity.

3. Materials and Methods

The symmetric key algorithm known as the Advanced Encryption Standard (AES) is more secure than other techniques. Because AES uses different key sizes, security is increased. AES is a block cipher that uses 128 bits for block sizes. This plaintext is 128 bits in size, but the key size can be changed to any value between 128 and 192 or 256 bits, which is significantly larger than DES. A round of operations is a collection of operations that are performed a certain number of times.

AES Bits	Key Size	Block Size	Rounds
128	128	128	10
192	192	128	12
256	256	128	14

Table 2: AES parameters

The length of the key can affect how many rounds there are. The longer the key and the more rounds used, the higher the level of security provided by AES. AES parameters are shown table 2. The number of 32-bit words in the algorithm's key is denoted by the symbol N_k. Depending on the key size being utilized (128 bits, 192 bits, or 256 bits), N_k accepts integers of 4, 6, or 8. The amount of AES rounds required depends on the key size that we utilize in our method. The value of "N_k," or the number of 32-bit words in the key, is what essentially determines the symbol N_r, or "number of cycles" in the algorithm. The number of cycles is denoted by N_r, with N_r equal to 10 for $N_k = 4$; 12 for $N_k = 6$; and 14 for $N_k = 8$ [6].

There are ten rounds, with a distinct, new key being used at each stage. These unique keys are produced by the AES key extension module. where the N_r keys will be generated using the original key. Each AES round basically consists of four processes: sub-bytes, mixing columns, moving rows, and adding round key shown in figure 1.

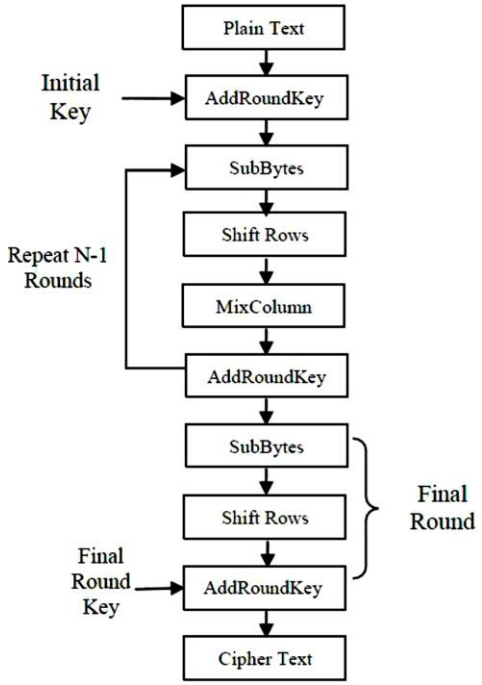

Figure 1: AES encryption

In the matching round, each distinct secret key generated by key expansion will be used. The given text is converted to hexadecimal first, and then it is made into a 4x4 state matrix. All operations are applied to this matrix. After that, each round matrix has a text format built for it. The AES method's input block, output block, and state array are all 128 bits long [3].

Sub-bytes: Every byte is impacted differently by the non-linear byte substitution. Here, each byte that is already present in the state array is changed to a new one. S-BOX is used to derive the data that needs to be replaced with the original data in the state matrix. This record contains pre-programmed information. The exact number for a specific piece of data that needs to be replaced is found using one byte in the table's column, and the second byte is used in the row. In the state matrices, the recently downloaded data from the S-box is replaced with the previous data by choosing the relevant row and column. Until all of the data in the state matrix is changed, these steps are repeated.

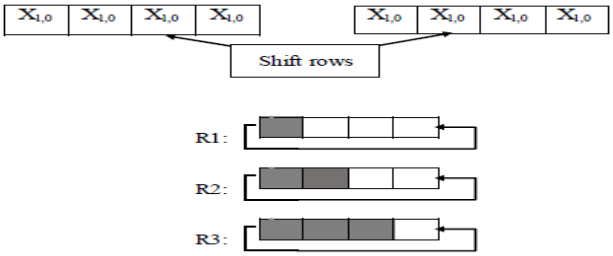

Figure 2: Shifting operation

Shift rows: During this process, the data in the state matrix cycle to the left. The location of the data in the state matrix determines how many shifts must be completed. It is completed in the matrix's rows. The data in the matrix's

final column is shifted cyclically once for row one. Two shifts are performed in the same matrix for row two. Since we are utilizing an index starting at "0," the first row in this process stays the same throughout all rounds. Shifting operation is shown above in the figure 2.

Mix-col: At this stage, the input is a matrix with shifted data in its rows. After that, this matrix is multiplied by another matrix made up of predefined values that is produced by using a common polynomial, a(X) [8]. With the exception of performing the XOR operation after multiplying the appropriate data in a row by the full column, as in matrix multiplication, rather than the addition operation, the matrix multiplication is essentially the same as conventional matrix multiplication.

Add round key: The data in the resultant state matrix from the preceding method is XORed with each of the state matrix's columns because the key in this stage is also 128 bits in size. In round zero of the encryption process, the first key is added. Every round requires a different key, and these unique keys are created from the initial key used in round 0. The number of rounds in the algorithm—which is completed in the key expansion module—determines the number of keys generated in this set.

Key expansion: Using a key expansion technique, AES creates a collection of round keys that are used for encryption and decryption. The first key is the input for the algorithm that generates the keys. The key expansion method, where N is the number of rounds in the AES algorithm, yields a total of (N+1) 128-bit round keys based on a sequence of modifications made to the initial key. The key expansion technique in Verilog HDL can be implemented with combinational and sequential logic circuits. The 128-bit initial key, which is kept in a register, is the input used by the key expansion algorithm. The round keys are then produced by the key expansion algorithm by executing many modifications on the original.

3.1 Secured algorithm

One of AES's advantages is that different key sizes can be used. Security can be improved by making it harder to breach by increasing the key size. Previously, 128-bit AES keys were utilized. The most common key size for increased security is Here, a low-power, less sophisticated AES with a 256-key capacity is used. Every round of operations is comparable to AES128. There will be 14 rounds of operations for AES256, depending on how the size of the key number of operations varies.

Key expansion produce round keys corresponding to the number of round. Transformations occur in key expansion is substitution and rotation. Then a random constant is xored. figure 3 shows the key expansion algorithm. Since the plaintext is 128 bits and the key size is 256, the key functions as a block of two 128 bit keys.

All five operations the first round operation, subbyte, shift rows, mix columns, and add round key operations—are included in the first round. Four operations—subbyte, shift rows, mix columns, and add round key operations—are included in the rounds two through thirteen.

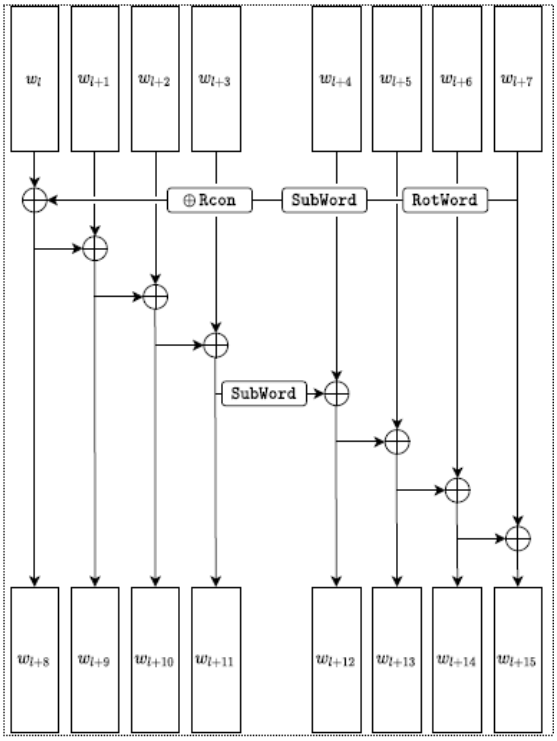

Figure 3: Key expansion for AES256

Additionally, there are three operations in the final 14th round: subbyte, shift rows, and add round key operations shown below in figure 4.

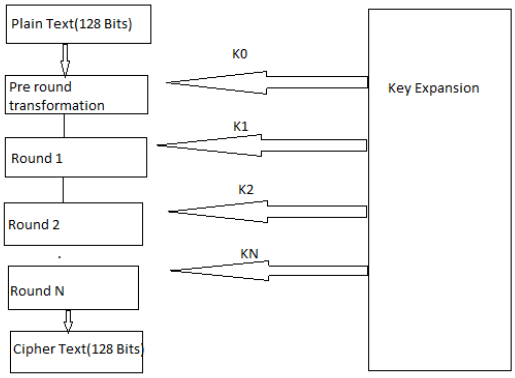

Figure 4: AES256 architecture

AES formerly used sbox of LUT, a preset matrix that is far less safe. Because AES used composite field arithmetic [7] Sbox has complicated signal channels that could result in high power consumption. Therefore, it would be best to employ composite field arithmetic, carefully dividing the composite Sbox into two level logic and converting it to PPRM (Positive Polarity Reed Muller) form. Pre-inversion, inversion, and post-inversion stages are the three distinct stages of the Sub-Bytes transformation at which these three stages of operation occur. An array of AND and XOR

gates can be used to implement all three steps. Just one of the outputs from the first stage needs to be calculated in the second stage; the output will be fed into the third stage [9-11].

4. Result and Discussion

Simulation results of AES256 algorithm and AES128 algorithm shown below in figure 5 (a) and (b) respectively.

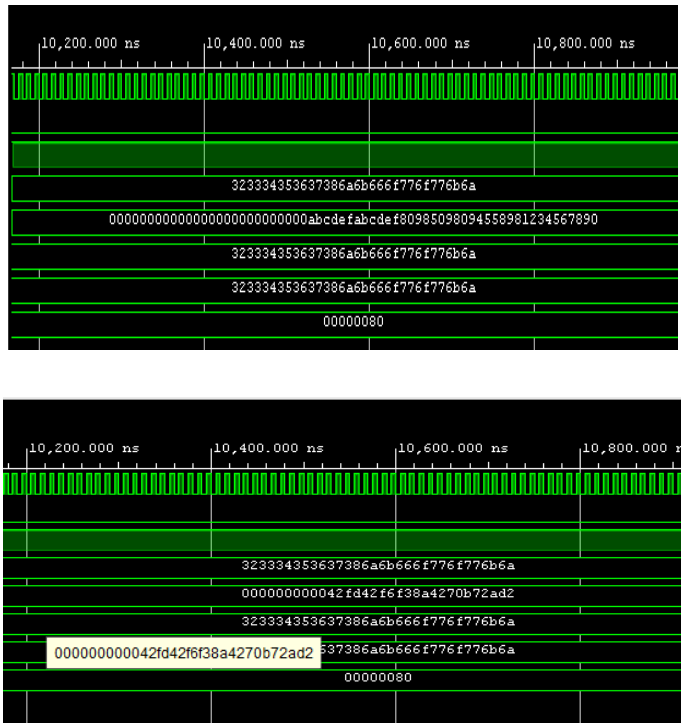

Figure 5: Simulated results (a)AES256 (b)AES128

By upgrading to AES256 and implementing the ideas lead to reduction in power, hardware complexity is not increased as well as we achieve constant delay. Power comparison is shown below in table 3.

	Area	Power	Delay
AES128	4635	236.869W	6.548
AES256	4636	211.983W	6.548

Table 3: Power analysis

5. Conclusions

The strongest degree of encryption is offered by the 256-bit key since it is the longest. A hacker would have to attempt 2256 distinct combinations with a 256-bit key in order to make sure they have the correct one. AES 256 is the most secure implementation of AES since the encryption becomes more difficult the more rounds there are. It should

be remembered that increasing performance requirements accompany longer keys and more rounds.Despite of the increase in rounds of operations due to increase in key size area or hardware complexity is not increased by using PPRM Sbox and sharing common resources. Also achieved lower power with constant area and delay by acquiring more secure algorithm as well.

6. References

[1] Nadeem H, "A performance comparison of data encryption algorithms", IEEE, 2005.

[2] B. Nageswara Rao, D. Tejaswi; K. Amrutha Varshini; K. Phani Shankar; B. Prasanth, "Design Of Modified AES Algorithm For Data Security", International Journal For Technological Research In Engineering, 2017.

[3] Nupur D. Vaidya; Yogesh; Suryawanshi; Manish Chavan, "Design for enhancing the performance of Advance Encryption Standard algorithm VHDL", International Conference on Green Engineering and Technologies (IC-GET), IEEE, 2016.

[4] V. H. Soumya; Mahesh B. Neelagar; K. V. Kumaraswamy, "Designing of AES Algorithm using Verilog", International Conference for Convergence in Technology (I2CT), IEEE, 2018.

[5] GuanLi Peng; Song Bai Zhu, "FPGA Implementation of AES Encryption Optimization Algorithm", IEEE, 2021.

[6] Keshav Kumar; K. Ramkumar; Amanpreet Kaur, "A Design Implementation and Comparative Analysis of Advanced Encryption Standard (AES) Algorithm on FPGA", International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), IEEE, 2020.

[7] M. Rajeswara Rao; Dr. R. K. Sharma, "FPGA Implementation of combined S box and Inv S box of AES", International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, 2017.

[8] Nalini C. Iyer; Deepa; P.V. Anandmohan; D.V. Poornaiah, "Mix/Inv MixColumn decomposition and resource sharing in AES", IEEE, 2010.

[9] Yulin Zhang; Xinggang Wang, "Pipelined implementation of AES encryption based on FPGA", IEEE International Conference on Information Theory and Information Security, 2010.

[10] Sumio Morioka and Akashi Satoh, "An Optimized S-Box Circuit Architecture for Low Power AES Design", IBM Research, Tokyo Research Laboratory, IBM Japan Ltd.

[11] Taosong Zhao, Hiroki Nishikawa, Xiangbo Kong, Hiroyuki Tomiyama, "Design and Evaluation of AES Encryption Circuits with Various S-Box Implementations", ATAIT 2023: The 5th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2023).