Partially Reconfigurable AES Algorithm with Dynamic Row Swapping in Run Time

Boppana MuraliKrishna[1], Mushtaq Hussain Shaik[2], Narindi Sai Chaitanya Kumar[3], Nikhila Chirumamilla[1],

Abstract. Cryptography is the method of gaining protection by encoding the original information, ensuring that only the intended recipients can interpret and process it. Any cryptographic algorithm has two important functions, encryption and decryption, which are complementary processes used for data integrity. Advanced Encryption Standard (AES) is a cryptographic algorithm that is one of the most important among other algorithms because it has keys of different sizes and a 128-bit input. A partial reconfiguration is applied to the AES algorithm to boost its ability to encrypt and decrypt vast amount of data with multiple permutations. This paper presents selective SubBytes row swapping through run time reconfiguration that uses partial reconfiguration in run time to swap rows in the first round depending on the mode selected. Modelsim is used to simulate the design, while Xilinx Vivado software is used to synthesize and implementation is carried on the Zynq FPGA (Field Programmable Gate Array) architecture. The results were verified using the Zed Board.

Keywords: Cryptography, Encryption, Decryption, AES, FPGA

1 Introduction

The primary goal of cryptography is to maintain data integrity, non-repudiation, credibility, and authenticity while maintaining confidentiality. In addition, Data Security is crucial in combating cyber threats such as unauthorized sensitive data access. There are two important and reputed data protection measures to consider. (a) Cryptography (b) Steganography. Steganography is the art of transmitting cryptographic hidden data through insecure networks. It is used via an insecure network to communicate, that only the administrator could acknowledge. Steganography is a technique of concealing the actual information with the help of utilizing duplicate information [1]. There are a number of different algorithms for providing data protection over insecure channels in communication. Data integrity is the key factor during the transmission of information between many individuals using such algorithms [2]. Data Encryption Standard (DES) and Triple DES are some cryptography algorithms other than Advanced Encryption Standard (AES) which are well known but, AES is the most prominent and widely used cryptography algorithm [16]. Encryption and decryption in cryptography should be in synchronization with one another based on the mechanisms of public and private key [3].

2. Literature Survey

The AES algorithm's functions include encryption and Decryption of scrambled data. Randomizing the information which is being transmitted through the channel by arranging the information into undecodable patterns. A fixed arrangement of is done in order to secure the data from any third-party operators is the process involved in encryption of data [4]. Using either a private key or public key the randomized information that is being transmitted through the channel can be recovered to its original structure [5]. The keys come in a variety of lengths and provide varying levels of protection depending on the length. At the transmitting end, keys are used to scramble the data. At the receiving end, keys are also needed to decrypt the cypher text. Cryptography is the method of combining the encoding and decoding of plain text.

2.1 Encryption

Encryption is the means by which information or data is secured. Data is transformed into encrypted data for this plain text as shown in figure 1. Cryptography is applied to preserve the information. Throughout this step, the data is converted into a specific code, such that sender may send it through the channel (primarily the internet) and no third party can access our data. Encryption typically consists of two subgroups: -Symmetric key encryption and Asymmetric key encryption. Symmetric key cryptography algorithm is by far the most widely employed Advanced Encryption Standard algorithm (AES).

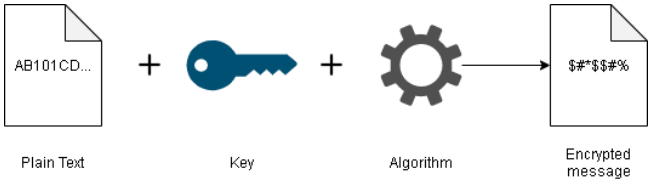


Fig. 1. Process of Encryption

2.2 Decryption

Decryption is the process of converting information which has already been rendered indecipherable down to its non - encrypted state via cryptography as depicted in figure 2. The machine collects and translates the incoherent data in decryption and translates it into images and text which are easy to comprehend not only by the reader but also by the machine. Decryption can be automatically or manually performed. It can also be accomplished with a collection of keys or passwords.

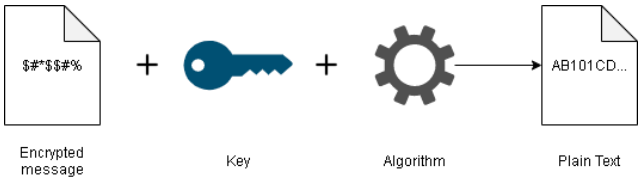


Fig.2. Process Decryption

2.3 Symmetric Encryption Technique

Encryption of the data is done by using a public key by both the sender and receiver on a predefined agreement. Encoded information is transmitted from the sender by using a key and several other operations. Receiver is able to decode the cipher after several arrangements using the same key [17]. Data is encrypted at the transmitter by undergoing several permutations and then decrypted by performing reverse permutations but using the same key as shown in figure 3.

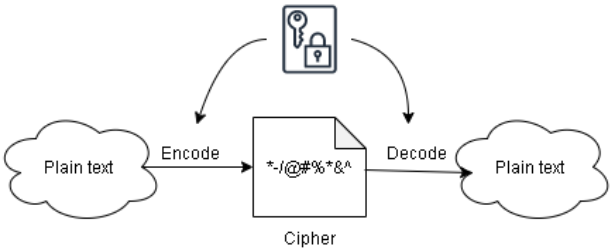


Fig.3. Encryption & Decryption using Symmetric Key

2.4 Asymmetric Encryption Technique

A mathematically related set of keys are used for encryption and decryption. Two different keys exist in asymmetric encryption technique, one is a private key and the other is a public key. The public key is used to encode the message in pre-cipher to produce cipher and the pre-cipher is encoded by using the private key as shown in figure 4 [6].

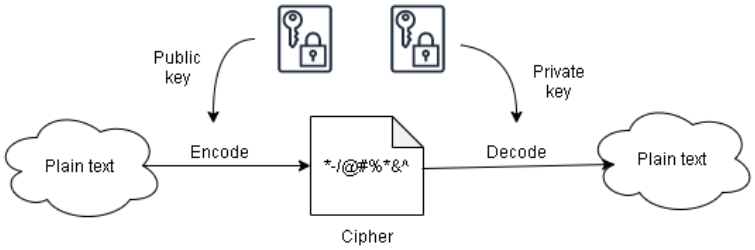


Fig.4. Encryption & Decryption using Asymmetric Key

2.5 Partial Reconfiguration

In Partial Reconfiguration a specific area on FPGA is modified without affecting any other application. Static PR and Dynamic PR are two types of functionality of a design flow, as illustrated in figure 5. Other parts in the FPGA are in shutdown mode while data is being loaded partially into the device. During the configuration of the device the Static PR is inactive. After reconfiguration is completed, the device returns back to its normal mode. Active PR is another term for Dynamic PR. When all other parts in an FPGA are still functioning, Dynamic PR has the capability to change the functionality of a specific part. Partial Reconfiguration is a process which allows the generation of partial bit files from the design flow. Figure 5 depicts static and dynamic PR [15].

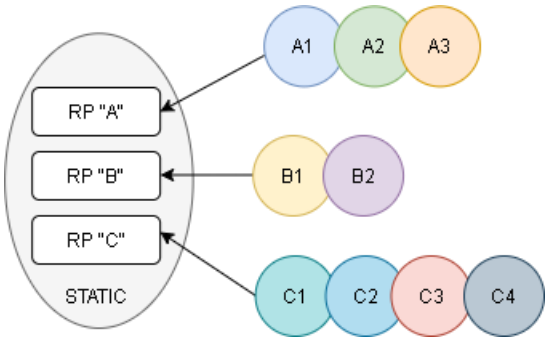


Fig.5. Encryption & Decryption using Asymmetric Key

3. Proposed Technique

Key generation algorithm generates values of keys in every round to execute round key operation. Intermediate key and Rcon values are present in key expansion. Depending on the bit size number of rounds to be performed during encryption are determined.10,12 or 14 rounds are performed corresponding to the size of the key,

which can be 128,192 and 256 respectively [7,18]. The key size for AES algorithm performed in this paper is consider as 128-bit. The changed ShiftRows operation with partial reconfiguration during run time is added to the AES algorithm after the sub bytes operation is completed. Depending on the mode selected in run time using partial reconfiguration, byte swapping occurs before moving rows during the first round. There are seven different cyphers created for each of the seven different modes. During each round, the plain text is transformed four times, adding to the AES algorithm's complexity. The input data is divided into a 4*4 matrix in the form of hexadecimal values extracted from plain text and key values. The dimensions of the matrix vary depending on the size of the key; for 192 and 256, the dimensions are 4*6 and 4*8, respectively. The algorithm will dynamically adjust depending on the size of the key since the number of rows remains constant at four. Column length in the matrix is not affected because in the calculations all the transformations performed are dependent on only the number of rows. Sub bytes, shift rows, mix columns and add round key are the four transformation operations present in the AES algorithm. In the first round, XOR operation is performed on the input data and the key bit in order to get a single key. The Sub Bytes and Inverse Sub Bytes tables as shown in figure 6 (a) and (b) have different values which are then added to the sequence [8]. The design flow is used to generate partial bit files in partial reconfiguration. Figure 7 depicts the procedure devised and followed to encrypt and decrypt in order to achieve the desired results. For 128 and 192-bit key lengths, the shift rows operation is performed by moving rows to the left depending on the row number. However, in this article, we use byte swapping before moving rows in the first round, which is achieved using partial reconfiguration during runtime.

Q

P	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

(a)

Q

P	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	54	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61

(b)

Fig.6. SubBytes and inverse SubBytes Tables (a), (b).

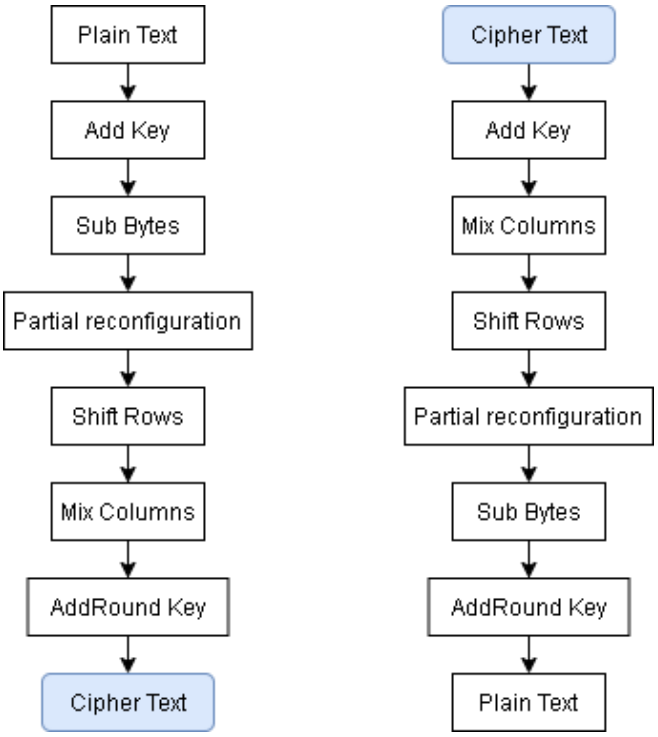


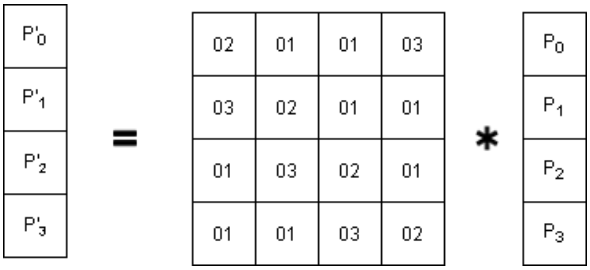
Fig.7. Encryption and Decryption Mechanism

In the mix columns step the entire data is modified rapidly in the algorithm. Encryption and decryption have completely different standard polynomial matrix. Implementation complexity is drastically reduced by using a code which has all its values dependent on a single .02 function. This function is as shown below.

X .02 = X<<1 (If left most bit of X is 0) = {X<<1} XOR 0001 1011 (If left most bit is 1)

Encryption involves certain polynomial functions that need to be calculated, they are represented below. AES algorithm is complicated and very efficient at manipulation because to produce a column in a matrix a total of 16 mathematical operations has to be done. General hexadecimal table can be noted in order to perform decryption which is exactly the reverse of the matrix in encryption. Using .02 function iteratively the simplification of the heavy calculations and decomplicating of the algorithm is done. At the end of each round keys are added which have been generated through key generation scheme. The mix column step is not involved in the final key for 10 rounds and 128-bit key.

X.03 = [X.02] XOR [X]



Po' = 02.P0 XOR 01. P! XOR 01.P2 XOR 03.P3

After adding byte swapping in the ShiftRows operation using partial reconfiguration in run time the generated cipher is transmitted across the channel.

3.1 Dynamic Shift Rows

Partial reconfiguration is used, instead of traditionally shifting rows; which is performing left shift operation depending on the row number. The ShiftRows process modifies the location of bytes Inside the state. Each row is rotated with different offsets to gain a new state as shown in figure 8.

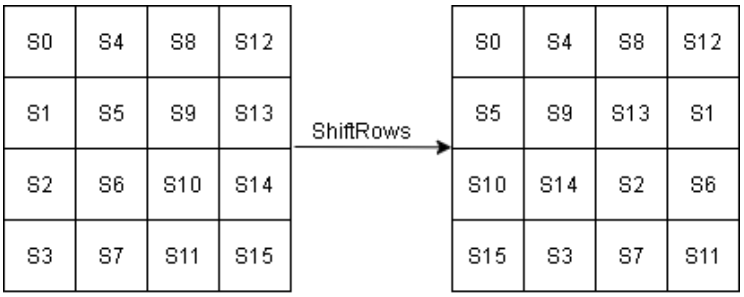


Fig.8. ShiftRows operation

The first row is constant as the row index is zero, thereafter left shift operation is performed on the second row and is shifted once. The third and last row are performed left shift twice and thrice respectively. This paper introduces slight modification that changed this approach of ShiftRows and used byte swapping to swap rows depending on the mode that is selected before ShiftRows function in the first round. Depending on the mode which we have selected the rows are swapped in real time as shown below in figure 9. This is done with the help of partial reconfiguration in run time.

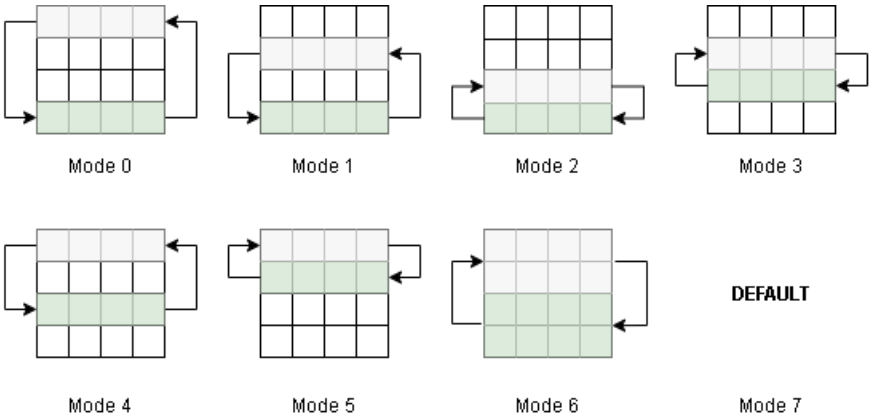


Fig.9. Row and Column swapping mechanism

Mode zero till mode six rows (0,3), (1,3), (2,3), (1,2), (0,2), (0,1), ((0,1), (2,3)) are swapped respectively. And Round seven has no byte swapping and is specified as default mode.

The default mode follows traditional AES encryption algorithm. Seven different ciphers are generated using seven different modes of operation. By doing this the data is scrambled randomly and makes harder for any third-party to access or crack our cipher [19,20].

3.2 MixColumn Operation

Each consecutive four bytes of state bytes are mixed by the MixColumns operation to receive four new bytes as illustrated in figure 10.

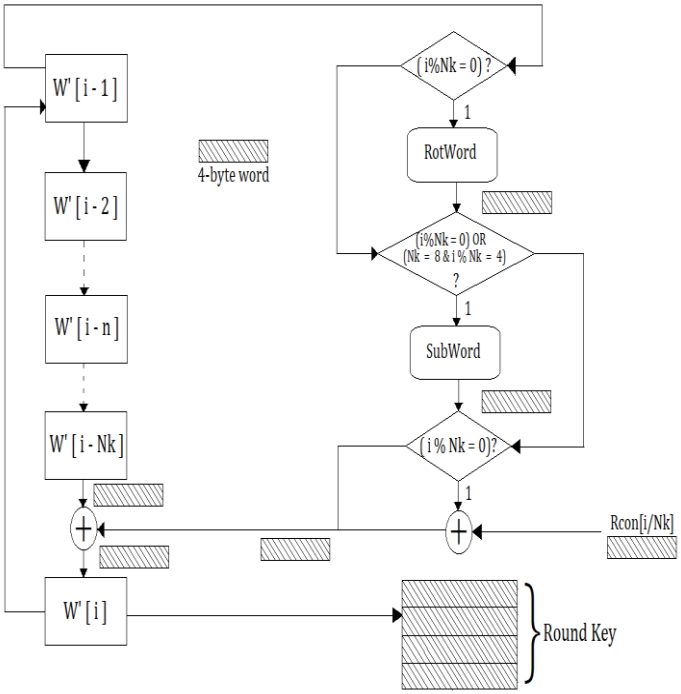


Fig.10. MixColumns Operation

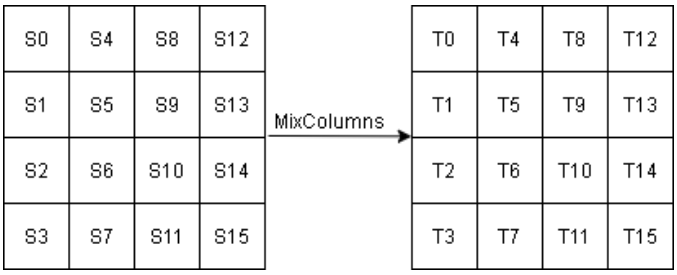


Fig.11. MixColumns State matrix

Each consecutive four is represented by S_i, S_{i+1}, S_{i+2} and S_{i+3} Bytes, with I ϵ {0,4,8,12}. The four bytes, then, are transformed by

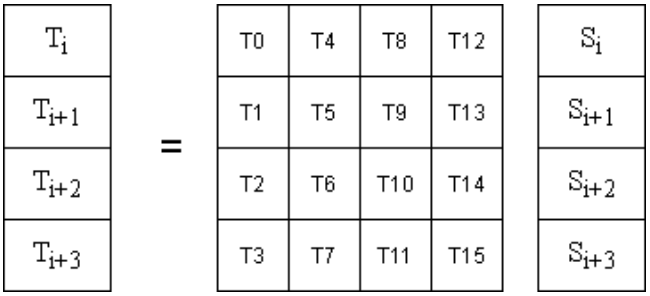


Fig.12. Equation for mix columns operation

Every constant matrix entry in the above equation shown in figure 12 corresponds to GF (2^8), So the equation in figure 12 is a multiplication of the matrix-vector over GF (2^8).

3.3 AddRoundKey Operation

The state matrix is Xored with the subkey in the AddRoundKey stage. Every round, a subkey is extracted from the main key using the Rijndael key schedule; each subkey will be the same size as the state. As shown in figure 13, the subkey is added using bitwise XOR by combining each byte of the state with the subkey's equivalent byte.

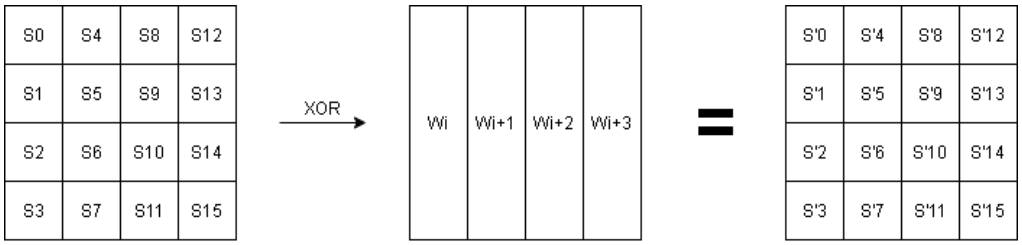


Fig.13. AddRoundKey Operation

3.4 Key Scheduling

Key scheduling or key expansion is a method of receiving the expected amount of subkeys from the main secret key for each round. Each subkey is reliant on a subkey from the preceding one. It's a very important operation, because there should be no similarity between any subkeys. It includes word rotating, word shifting, addition of a value constant, EXOR operation [9-14].

4. FPGA Implementation

Application-specific integrated circuits (ASICs) or processors are especially well suited to FPGA prototyping. An FPGA can be reprogrammed before the ASIC or processor design is finalized and bug-free, at which point the final ASIC can be manufactured. FPGAs are used by Intel to prototype new chips. Partial reconfiguration is changing the area of an FPGA without affecting other applications. Multiple modules can be simulated without disrupting the flow. Partially activated modules can be reconfigured using partial reconfiguration. The ARM Vertex a9 processor is used to monitor the progress of dynamically formed Verilog code. In many applications, Vertex a9 has improved peak output and efficiency while consuming less power. Hence, proving its mettle. The vertex a9 processor was used to achieve the desired performance. The processor's maximum performance resolution is 330 bits. The AES cypher needs 512 bits of storage to transmit encrypted data, with a byte swapping mechanism that uses partial reconfiguration in runtime. 256-bit is the size of two sub modules which are divided and is well within the range of the processor bit size capability.

5 Analysis of Result

Cipher developed using the AES data encryption with hexadecimal input message signal, Sub Bytes, Cipher output for the Mode (0,1,7) that have been selected are all

shown in the Fig. 14,15,16. Figure 17 shows the physical view of AES algorithm with modes functionality added to it on Zynq FPGA architecture.

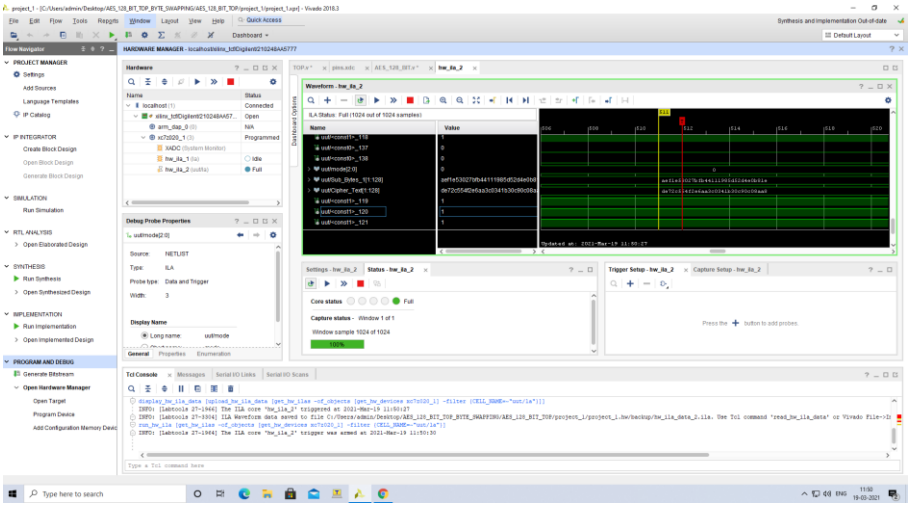


Fig.14. Cipher generated in mode 0.

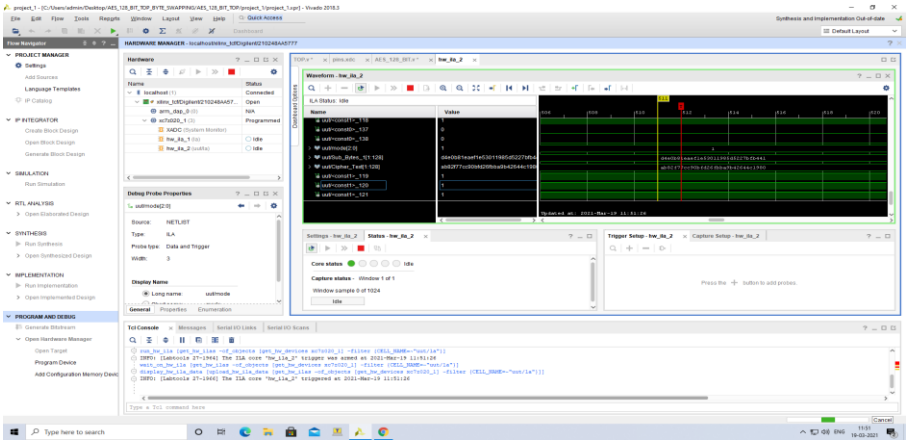


Fig.15. Cipher generated in mode 1.

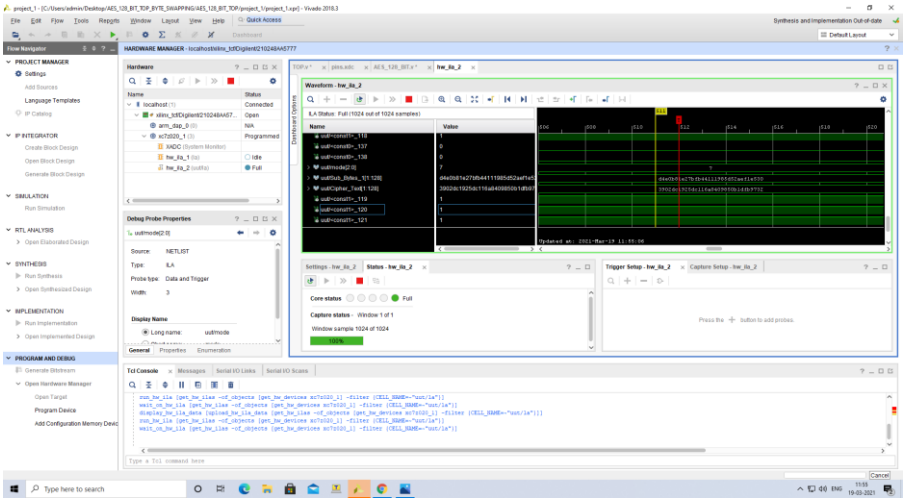


Fig.16. Cipher generated in mode 7 or default mode.

Fig.17. Physical view AES algorithm on Zync FPGA architecture.

Mode 0:
Sub Bytes: aef1e53027bfb4111985d52d4e0b81e
Cypher: de72c554f2e6aa3c0341b30c90c08aa8

Mode 1:
Sub Bytes: d4e0b81eaef1e53011985d5227bfb441

Cypher: ab82f77cc90bfd26fbba9b42644c1980

Mode 2:
Sub Bytes: d4e0b81e27bfb441aef1e53011985d52
Cypher: ea9d33568a270bf6bf6bf142d42e1471

Mode 3:
Sub Bytes: d4e0b81e11985d5227bfb441aef1e530
Cypher: 3aed055d6e46b4c8b99128642e97a384

Mode 4:
Sub Bytes: 11985d5227bfb441d4e0b81eaef1e530
Cypher: 00b62919e0421d5d65952fe23becd763

Mode 5:
Sub Bytes: 27bfb441d4e0b81e11985d52aef1e530
Cypher: 366a04d8006a9e0346a006e0003306a6

Mode 6:
Sub Bytes: 11985d52aef1e530d4e0b81e27bfb441
Cypher: c1b112ee30b3d9c2a508652ed12c7d2e

Mode 7:
Sub Bytes: d4e0b81e27bfb44111985d52aef1e530
Cypher: 3902d61925dc116a8409850b1dfb9732

Conclusion

AES algorithm has been simulated and implemented and the cypher has been generated for different modes by using a byte swapping mechanism to swap rows depending on the mode that is being selected in run time. The security levels have been bolstered by manipulating the bytes in round one using partial reconfiguration. Byte swapping is applied during the shift rows operation using partial reconfiguration in run time while transmitting the cipher and the round key through the channel in the first round. Depending on the time stamp the receiver can decipher the code by assessing the mode in which the cipher has been generated. The diversified approach of adding partial reconfiguration technique and byte swapping to the AES algorithm bolstered its security capabilities. Future work includes the decryption of data using this method. Zynq FPGA architecture was used for the testing of results.

References

[1] Advances in Intelligent Systems and Computing 340 J. K. Mandal, Suresh Chandra Satapathy, Manas Kumar Sanyal, ParthaPratimSarkar,

AnirbanMukhopadhyay (eds.) - “Information Systems Design and Intelligent Applications” pg 207 to 215 in the year 2015.
[2] Sreeja C.S, Mohammed Misbahuddin, Mohammed Hashim N.P,” DNA for Information Security: A Survey on DNA Computing and a Pseudo DNA Method Based on Central Dogma of Molecular Biology”, IEEE 2014.
[3] Yaser Jararweh, AES-512: 512-Bit Advanced Encryption Standard Algorithm Design and Evaluation, IEEE, 2011.
[4] Manjesh.K.N, R K Karunavathi, Secured High throughput implementation of AES Algorithm, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, May 2013 ISSN: 2277 128X.
[5] Yaser Jararweh, AES-512: 512-Bit Advanced Encryption Standard Algorithm Design and Evaluation, IEEE, 2011.
[6] Chen-Hsing Wang, Chieh-Lin Chuang, and Cheng-Wen Wu; An Efficient Multimode Multiplier Supporting AES and Fundamental Operations of Public-Key Cryptosystems, IEEE Transactions on Very Large-Scale Integration (VLSI) systems, vol. 18, No. 4, April 2010.
[7] Mukta Sharma, Abdul Wahid Ali“comparative analysis of npn algorithm & des algorithm”, 2015 international conference on computing, communication & automation,15-16 may 2015.
[8] SoufianeOukili, SeddikBri “FPGA implementation of Data Encryption Standard using time variable permutations”, International Conference on Microelectronics, 20-23 Dec. 2015, pp 126 – 129.
[9] Abuelyman ES, Alsehibani AAS (2008) An optimized implementation of the s-box using residue of prime numbers. Int J Computer Science and Network Security 8(4):304–309
[10] Itoh T, Tsujii S (1988) A fast algorithm for computing multiplicative inverses in GF(2m) using normal bases. Inf Comput 78(3):171–177
[11] Good T, Benaissa M (2006) Very small FPGA application specific instruction processor for AES. IEEE Trans Circuits Syst I Regul Pap 53(7):1477–1486
[12] Good T, Benaissa M (2010) 692-nW advanced encryption standard (AES) on a 0.13-m CMOS. IEEE Trans Very Large-Scale Integration VLSI Syst 18(12):1753–1757
[13] Huang J, Lai X (2012) Transposition of AES key schedule. Int Assoc Cryptal Res 260:1–13
[14] Paul A, Victoire TAA, Jeyakumar AE (2003) Particle swarm approach for retiming in VLSI. In: IEEE 46th Midwest symposium on 3 circuits and systems, 2003, pp 1532–1535.
[15] M. Pranav, Archana K Rajan, “DES security enhancement with dynamic permutation”, International Conference on Applied and Theoretical Computing and Communication Technology,29-31 Oct. 2015, pp 6-11.
[16] US Nat’l Inst. of Standards and Technology, “Federal Information Processing Standards Publication 197—Announcing the ADVANCED ENCRYPTION STANDARD (AES),” 2001, http:// csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
[17] W. Stallings, CRYPTOGRAPHY AND NETWORK SECURITY PRINCIPLES AND PRACTICE, Fifth Edit., vol. 139, no. 3. Pearson Education, Inc, 2011.

[18] Daemen, J. and Rijmen, V. (2002) The Design of Rijndael: AES – the Advanced Encryption Standard. Springer, Berlin.
[19] Tausif Anwar, Abhishek Kumar, Sanchita Paul; DNA Cryptography Based on Symmetric Key Exchange, International Journal of Engineering and Technology (IJET).
[20] Prasoon Raghav, Rahul Kumar, Rajat Parashar, Securing Data in Cloud Using AES Algorithm, ISSN 2321 3361, IJESC.