IJERST

International Journal of Engineering Research and Science & Technology

ISSN : 2319-5991
Vol. 3, No. 4
November 2014

www.ijerst.com

Email: editorijerst@gmail.com or editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
Vol. 3, No. 4, November 2014
© 2014 IJERST. All Rights Reserved

Research Paper

REALIZATION OF SECURITY-ENABLED COMMUNICATION WITH LOW COST FLEXIBLE ARCHITECTURE SUPPORTING SYMMETRIC CRYPTOGRAPHY

Haneesha Alla[1]* and N Suresh Babu[1]

Corresponding Author: **Haneesha Alla ✉ haneesha99@gmail.com*

Data encryption (cryptography) is utilized in various applications and environments. The specific utilization of encryption and the implementation of the AES will be based on many factors particular to the computer system and its associated components. Communication security provides protection to data by enciphering it at the transmitting point and deciphering it at the receiving point. File security provides protection to data by enciphering it when it is recorded on a storage medium and deciphering it when it is read back from the storage medium. In the proposed design the security method uses symmetric Cryptography technique which provides same keys to sender and receiver to transfer the information for reducing the design complexity. The transferred information is stored in the memory unit for further using it or for further processing using low cost buffer element. Transmission cables are used to provide communication between the connected devices. Control logic is used to provide the patters for transmission through crypto unit.

Keywords: Cryptography, Memory, Control Unit, Symmetric

INTRODUCTION

AES is short for Advanced Encryption Standard and is a United States encryption standard defined in Federal Information Processing Standard (FIPS) 192, published in November 2001. It was ratified as a federal standard in May 2002. AES is the most recent of the four current algorithms approved for federal us in the United States. One should not compare AES with RSA, another standard algorithm, as RSA is a different category of algorithm. Bulk encryption of information itself is seldom performed with RSA.RSA is used to transfer other encryption keys for use by AES for example, and for digital signatures.

AES is a symmetric encryption algorithm processing data in block of 128 bits. A bit can take the values zero and one, in effect a binary digit with two possible values as opposed to

[1] M.Tech. Student, Department of ECE, Chirala Engineering College, Chirala 523155, Prakasam Dt., AP.
[2] Associate Professor, Department of ECE, Chirala Engineering College, Chirala 523155, Prakasam Dt., AP.

decimal digits, which can take one of 10 values. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size. AES is symmetric since the same key is used for encryption and the reverse transformation, decryption. The only secret necessary to keep for security is the key. AES may configured to use different key-lengths, the standard defines 3 lengths and the resulting algorithms are named AES-128, AES-192 and AES-256 respectively to indicate the length in bits of the key. Each additional bit in the key effectively doubles the strength of the algorithm, when defined as the time necessary for an attacker to stage a brute force attack, i.e. an exhaustive search of all possible key combinations in order to find the right one.

AES is founded on solid and well-published mathematical ground, and appears to resist all known attacks well. There's a strong indication that in fact no back-door or known weakness sexists since it has been published for a long time, has been the subject of intense scrutiny by researchers all over the world, and such enormous amounts of economic value and information is already successfully protected by AES. There are no unknown factors in its design, and it was developed by Belgian researchers in Belgium therefore voiding the conspiracy theories sometimes voiced concerning an encryption standard developed by a United States government agency. AES may, as all algorithms, be used in different ways to perform encryption. Different methods are suitable for different situations. It is vital that the correct method is applied in the correct manner for each and every situation, or the result may well be insecure even if AES as such is secure. It is very easy to implement a system using AES as its encryption algorithm, but much more skill and experience is required to do it in the right way for a given situation. No more than a hammer and a saw will make anyone a good carpenter, will AES make a system secure by itself. To describe exactly how to apply AES for varying purposes is very much out of scope for this short introduction.

Encryption with AES is based on a secret key with 128, 192 or 256 bits. But if the key is easy to guess it doesn't matter if AES is secure, so it is as critically vital to use good and strong keys as it is to apply AES properly. A creating good and strong key are a surprisingly difficult problem and requires careful design when done with a computer. The challenge is that computers are notoriously deterministic, but what is required of a good and strong key is the opposite unpredictability and randomness. Keys derived into a fixed length suitable for the encryption algorithm from passwords or pass phrases typed by a human will seldom correspond to 128 bits much less 256. To even approach 128—bit equivalence in a pass phrase, at least 10 typical passwords of the kind frequently used in day-to-day work are needed. Weak keys can be somewhat strengthened by special techniques by adding computationally intensive steps which increase the amount of computation necessary to break it. The risks of incorrect usage, implementation and weak keys are in no way unique for AES; these are shared by all encryption algorithms. Provided that the implementation is correct, the security provided reduces to a relatively simple question about how many bits the chosen key, password or pass phrase really corresponds to. Unfortunately this estimate is somewhat difficult to calculate, when the key is not generated by a true random generator.

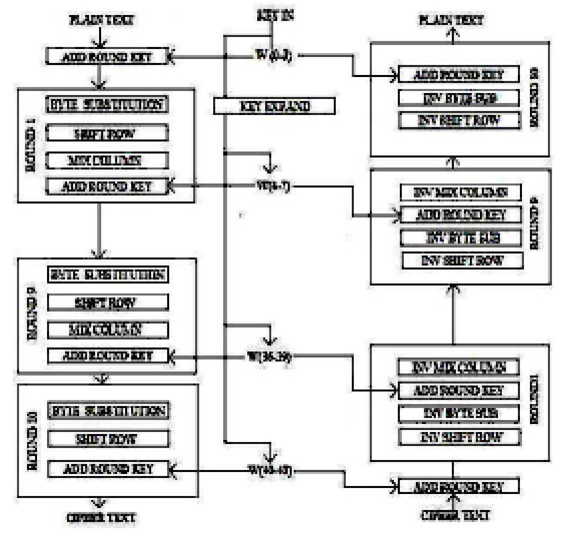
Security is not an absolute; it's a relation between time and cost. Any question about the security of encryption should be posed in terms of how long time, and how high cost will it take an attacker to find a key? Currently, there are speculations that military intelligence services possibly have the technical and economic means to attack keys equivalent to about 90 bits, although no civilian researcher has actually seen or reported of such a capability. Actual and demonstrated systems today, within the bounds of a commercial budget of about 1 million dollars can handle key lengths of about 70 bits. An aggressive estimate on the rate of technological progress is to assume that technology will double the speed of computing devices every year at an unchanged cost. If correct, 128-bit keys would be in theory be in range of a military budget within 30-40 years. An illustration of the current status for AES is given by the following example, where we assume an attacker with the capability to build or purchase a system that tries keys at the rate of one billion keys per second. This is at least 1 000 times faster than the fasted personal computer in 2004. Under this assumption, the attacker will need about 10000000000000000000000 years to try all possible keys for the weakest version, AES-128. The key length should thus be chosen after deciding for how long security is required, and what the cost must be to brute force a secret key. In some military circumstances a few hours or days security is sufficient – after that the war or the mission is completed and the information uninteresting and without value. In other cases a lifetime may not be long enough.

ADVANCED ENCRYPTION STANDARDS (AES)

AES is a specification for the encryption of electronic data. It has been adopted by the US. government and is now used worldwide. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.

The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

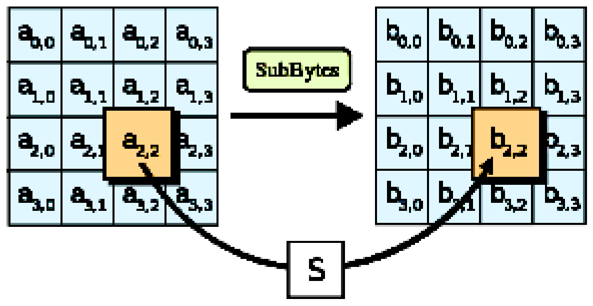
Figure 1: Encryption and Decryption Architecture



The Sub Bytes Step

In the Sub Bytes step, each byte in the matrix is updated using an 8-bit substitution box,. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse, known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen

Figure 2: Sub Bytes Information

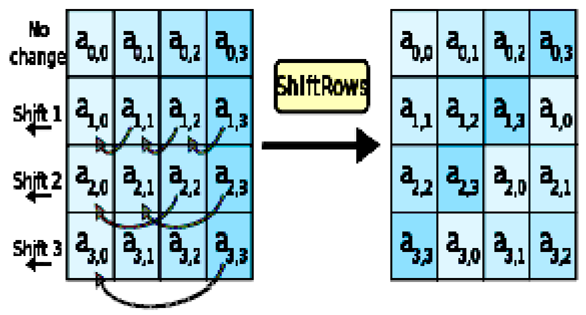


The S-box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points.

Shift Rows Step

The Shift Rows step operates on the rows of the state, it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For the block of size 128 bits and 192 bits the shifting pattern is the same. In this way, each column of the output state of the Shift Rows step is composed of bytes from each column of the input state.

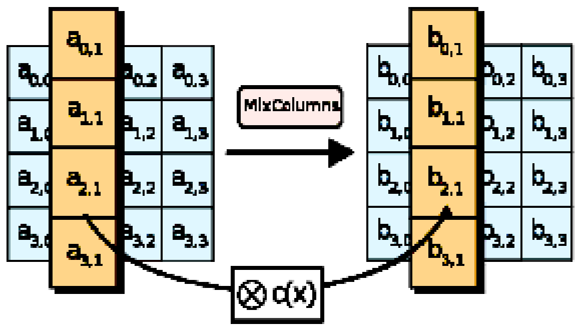
Figure 3: Shift Rows Information



Mix Columns Step

In the Mix Columns step, the four bytes of each column of the state are combined using an invertible linear transformation. The Mix Columns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes.

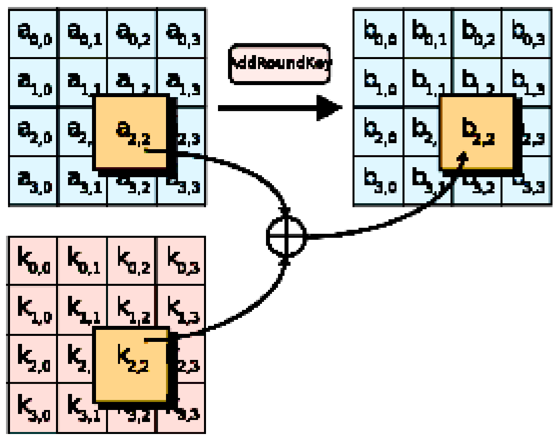
Figure 4: Mix Columns Information



The Add Round Key Step

In the Add Round Key step, the sub key is combined with the state. For each round, a sub key is derived from the main key using key schedule. Each sub key is the same size as the state. The sub key is added by combining each byte of the state with the corresponding byte of the sub key using bitwise XOR.

Figure 5: Add Around Key Information



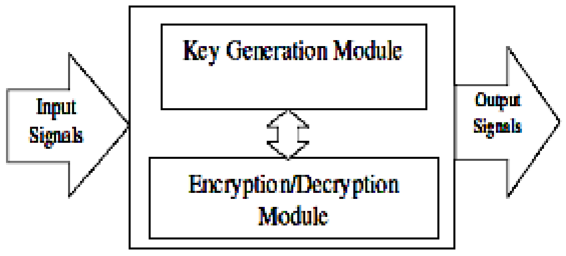
PROPOSED MODEL

The proposed architecture is designed to get less power consumption and lesser area by mapping all the four Logical functions of AES to LUTs, ROMs and Block RAMs. The proposed architecture has three parts

1. Key Generation Module
2. Encryption Module
3. Decryption Module.

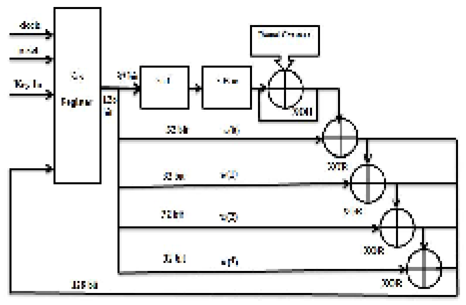
The AES encryption and decryption core unit contains key generation module as a common unit. This module gives necessary key expansion for both encryption and decryption functions. Fig.3 presents the block diagram of AES encryption and decryption with Key Generation Module as a common unit. The key generation module consists of key register of 128 bits, S-Box and XOR gates for bitwise XOR operation.

Figure 6: AES Encryption and Decryption Unit Block Diagram



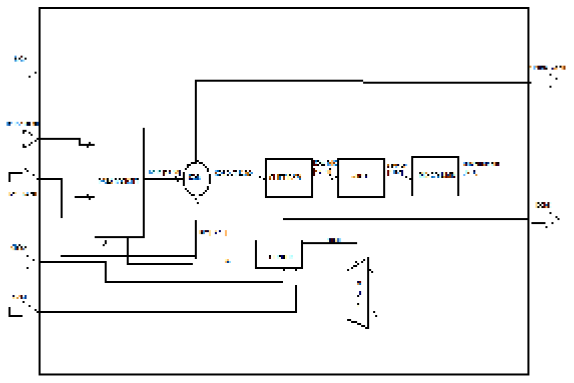
It is designed to produce round keys on each positive edge of the clock, when it is enabled. However in the proposed work, the key generation architecture does not require any hardware for shift operation and the port mapping between key register and S-Box is done according to the required shift. Hence the proposed work offers the advantage in area. Also in the proposed work the bits are rearranged on data path from register to S-Box and the round constant required for each rounds are stored in ROM and retrieved on each clock. Figure 3.2 represents proposed architecture of key generation unit.

Figure 7: Architecture of Key Generation Module



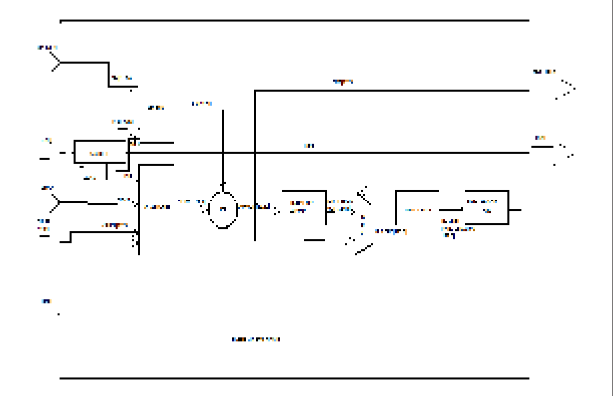
The encryption module takes 128 bit text to be encrypted and receives round key from key generation module to do each round of encryption. In the proposed work for reducing the hardware of entire architecture, the control unit of encryption module is not designed separately. The control unit of key generation module which is a 4-bit counter is designed to control the entire functioning of encryption module. The sharing of control unit by both encryption and round key generation gives unique advantage of reduction in hardware as compared to other implementations.

Figure 8: Encryption Module



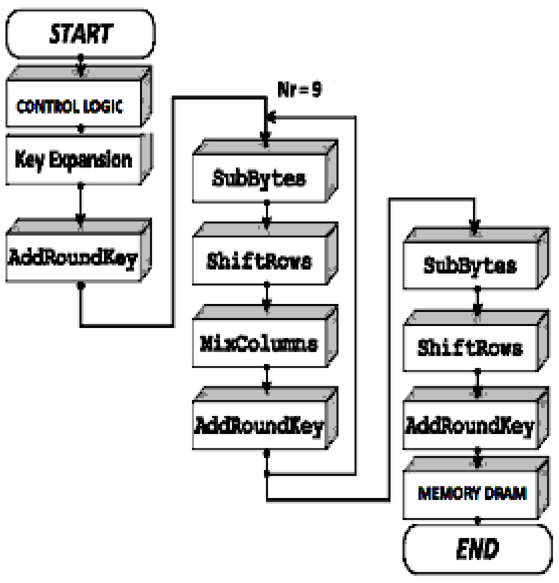
In the proposed work the Mix Column encryption hardware uses two of such ROM for Galois multiplication of '2' and '3' and for performing 4-Input XOR operation in Mix Column operation, the proposed design use 16 x 1 ROM with the result that Mix Column operation offers higher speed and uses minimum number of slices in the hardware (FPGA).

Figure 9: Decryption Module



The decryption unit also uses same design approach for the entire architecture to decrypt the given cipher back to original text. Inverse S-Box architecture uses the same design of S-Box. Entry of LUT is changed according to Inverse Sub Byte transformation. Mix Column operation is implemented using 256X8 ROM. 4-Input XOR operation is designed by 16x1 ROM. Architecture of Decryption module is same as encryption module with all complimentary functions of encryption. Decryption unit contains an extra register for storing Round Keys. Storing key is important since first round decryption use tenth round key and second round use ninth round key and so on.

Figure 10: Flow Diagram of the Proposed Design



The flow diagram describes the way of execution of the project. The above diagram explains the encryption process where as the decryption process is nothing but the reverse operation of the encryption. The control logic takes care about the synchronization and the initialization process. In the paper the memory module used to store the information after performing the decipher process is DRAM. Due to this there is no wastage of memory which will be initialized only on at the run time of the processor.

RESULTS

Figure 11: Waveform

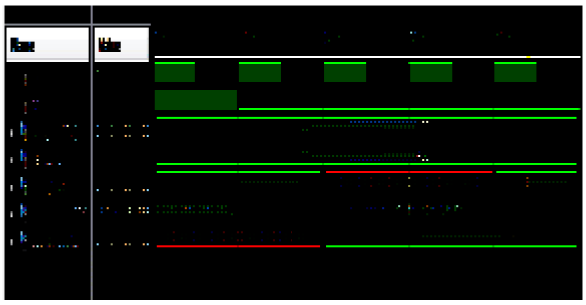


Figure 12: RTL Schematic View

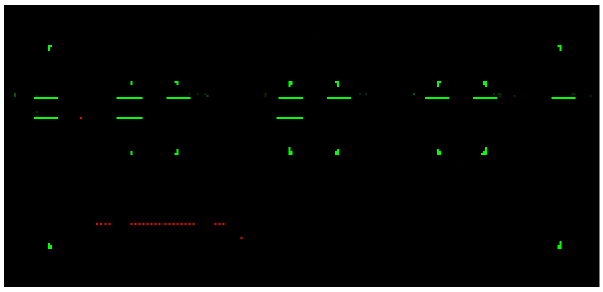


Figure 13: Technology View

CONCLUSION

In the proposed design the security method uses symmetric Cryptography technique which provides same keys to sender and receiver to transfer the information for reducing the design complexity. This proposed architecture Widely applied for computer and communication network applications, Information security applications, military, political and diplomatic fields. From the results it can be concluded that the area required to integrate the chip is less with the gate count of 20585 with reduced power consumption of 43.907mw. The functionality is verified using ISE simulator and the synthesis is carried out using XILINX ISE 12.3i.

REFERENCES

1. Hennessy J L and Patterson D A (2007), Computer Architecture: A Quantitative *Approach*, fourth ed., Morgan Kaufmann.
2. Morioka S and Satoh A , "A 10-gbps full-AES Crypto Design with a Twisted BDD s-Box Architecture," *IEEE Trans. Very Large Scale*.
3. Mukhopadhyay D and RoyChowdhury D (2005), "An Efficient end to End Design of Rijndael Cryptosystem in 0:18m CMOS," *Proc.18th Int'l Conf. VLSI Design,* pp. 405-410, Jan.
4. NIST (1999), "Data Encryption Standard (DES)," http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf, Oct.
5. NIST (2001), "Advanced Encryption Standard (AES)," http://csrc.nist.-gov/publications/fips/fips197/fips-197.pdf, Nov.
6. "Security-Enabled Near-Field Communication Tag with Flexible architecture Supporting Asymmetric Cryptography", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 21, No. 11.
7. Verbauwhede I, Schaumont P and Kuo H (2003), "Design and Performance Testing of a 2.29 gb/s Rijndael Processor," *IEEE J. Solid-State Circuits*, Vol. 38, No. 3, pp. 569-572.

ISSN 2319-5991