ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

Implementation of Cryptography Encryption Algorithm for Plane Text

BV Ramudu[1], Sanjeeva Polepaka[2]

Assoc Professor, MallaReddy Engineering College (Autonomous) Hyderabad, India[1,2]

Abstract: In this paper an efficient security architecture design and implementation of all candidates of AES encryption standards AES-128, AES-192 and AES-256 on the same hardware is proposed. AES algorithm proposed by NIST has been widely accepted as best cryptosystem for wireless communication security. The hardware implementation is useful in wireless security like military and mobile phones. This contribution investigates implementation of AES Encryption with regards to FPGA and VHDL.Optimized and synthesized VHDL code for AES-128, AES-192 and AES-256 for encryption of 128- bit data is implemented. Xilinx ISE 9.2i software is used for simulation. Each algorithm is tested with sample vectors provided by NIST output results are perfect with minimal delay. The proposed design consumes less power and area which is suitable battery driven mobile phones. Throughput reaches the value of 666.67 Mbps for encryption of 128- bit data with AES-128 key.

Keywords: Cryptography, Cipher, Reconfiguration, Encryption, Decryption.

1. INTRODUCTION

Cryptography is the science of information and communication security. Cryptography is the science of secret codes, enabling the confidentiality of communication through an insecure channel. It protects against unauthorized parties by preventing unauthorized alteration of use. Generally speaking, it uses an cryptographic system to transform a plaintext into a cipher text, using most of the time a key. One has to notice that there exist certain ciphers that don't need a key at all. A famous example is ROT13 (abbreviation from Rotation 13), a simple Caesar-cipher that obscures text by replacing each letter with the letter thirteen places down in the alphabet. Since our alphabet has 26 characters, it is enough to encrypt the cipher text again to retrieve the original message. Let me just mention briefly that there are secure public-key ciphers, like the famous and very secure Rivest-Shamir-Adleman (commonly called RSA) that uses a public key to encrypt a message and a secret key to decrypt it. Cryptography is a very important domain in computer science with many applications. The most famous example of cryptography is certainly the Enigma machine, the legendary cipher machine used by the German Third Reich to encrypt their messages, who's security breach ultimately led to the defeat of their submarine force. Before continuing, please read carefully the legal issues involving cryptography as in several countries even the domestic use of cryptography is prohibited: Cryptography has long been of interest to intelligence gathering agencies and law enforcement agencies. Because of its facilitation of privacy, and the diminution of privacy attendant on its prohibition, cryptography is also of considerable interest to civil rights supporters. Accordingly, there has been a history of controversial legal issues surrounding cryptography, especially since the advent of inexpensive computers has made possible widespread access to high quality cryptography. In some countries, even the domestic use of cryptography is, or has been, restricted. Until 1999, France significantly restricted the use of cryptography domestically. In China, a license is still required to use cryptography. Many countries have tight restrictions on the use of cryptography. Among the more restrictive are laws in Belarus, Kazakhstan, Mongolia, Pakistan, Russia, Singapore, Tunisia, Venezuela, and Vietnam.[31]

In the United States, cryptography is legal for domestic use, but there has been much conflict over legal issues related to cryptography. One particularly important issue has been the export of cryptography and cryptographic software and hardware. Because of the importance of cryptanalysis in World War II and an expectation that cryptography would continue to be important for national security, many western governments have, at some point, strictly regulated export of cryptography. After World War II, it was illegal in the US to sell or distribute encryption technology overseas; in fact, encryption was classified as a munitions, like tanks and nuclear weapons.[32] Until the advent of the personal computer and the Internet, this was not especially problematic. Good cryptography is indistinguishable from bad cryptography for nearly all users, and in any case, most of the cryptographic techniques generally available were slow and error prone whether good or bad. However, as the Internet grew and computers became more widely available, high quality encryption techniques became well-known around the globe. As a result, export controls came to be seen to be an impediment to commerce and to research.

1.1 Description of AES Algorithm:

Advanced Encryption Standard is the successor of Data Encryption Standard which was in use during the early 1977 to 1990. In DES encryption is based on a symmetric key algorithm that uses a 56-bit key. However by the mid 1990's, it was clear that the DES with 56-bit is insecure for many applications since the key is very small. Then it

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

was upgraded to Triple DES which was believed to be practically secure although there are theoretical attacks. Thus in Nov-26-2001 the FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION 197(FIPS 197) specifies an algorithm called Advanced Encryption Standard (AES). AES is based on the principle known as Substitution Permutation network (SP-network) which means there will be a series of linked mathematical operations in the block cipher algorithm.AES encrypts a data block of 128-bits which is fixed with three different key sizes 128,192,256 bits.

The operations are based on Rinjdael algorithm. The input of AES algorithm is 128-bit or 16 byte data which can be specified as a block. The basic unit of processing in the AES algorithm is a byte. All byte values in the AES algorithm will be presented as the concatenation of its individual bit values (0 or 1) between the braces in the order (b7, b6, b5, b4, b3, b2, b1, b0). These bytes are interpreted as finite field elements using a polynomial representation as follows

Algorithm	Key length (Nk words)	Block Size (Nb words)	Number of rounds (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

$b_7X^7+ b_6X^6+ b_5X^5+ b_4X^4+ b_3X^3+ b_2X^2+ b_1X+b_0 =$ $\sum_{i=0}^{7} X^i b_i$

Internally in AES algorithm operations are performed on a two-dimensional array of bytes called the state. The state consists of four rows of bytes, each containing Nb bytes, where Nb is the block length divided by 32 (4 for 128-bit key, 6 for 192-bit key, 8 for 256-bit key). Likewise the key length and number of rounds (iterations) differ from key to key as shown in table 1.
Figure1: Different keys and its attributes

AES Encryption:

Encryption is the process of converting the plain text into a format which is not easily readable and is called as cipher. The cipher is got by doing a series of mathematical operations iteratively.

a) Sub Bytes:

In this sub bytes step the data in the plain text is substituted by some pre-defined values from a substitution box. The substitution box is invertible.

b) Shift Rows:

In shift rows operation the rows in the 4×4 matrix is shifted to left r bits and r varies with the rows of the matrix(r=0 for row1, r=1 for row2, r=2 for row3, r=3 for row 4). This process is illustrated in fig 2.This has the effect of moving positions of lower positions in the row, while the lowest bytes wrap around to the top of the row.

c) Mix Columns:

Mix column is calculated using the below formula.

Here a0, a1, a2, a3 is calculated using the polynomials as below

$$a(x) = \{2\}x^3 + \{3\}x^2 + \{1\}x + \{1\}.$$

The mix column transformation operates on the state column by column, treating each column as a four term polynomial. The columns are considered as polynomials over GF (2^8) and multiplied modulo $x^4 + 1$ with a fixed polynomial a(x) which is got from the above formula. This can also written as a matrix multiplication

$$s'(x) = a(x)\ \ s(x)$$

d) Add Round Key:

In the add round key step the 128 bit data is xored with the sub key of the current round using the key expansion operation. The add round key is used in two different places one during the start that is when round r=0 and then during the other rounds that is when $1 \leq$ round$\leq$ Nr, where Nr is the maximum number of rounds. The formula to perform the add round key is S'(x) = S(x) R(x)

where S'(x) – state after adding round key ,S(x) – state before adding round key and R(x) – round key

e) Key Expansion:

The key expansion has three steps: Byte Substitution subword(),Rotation rotword() and Xor with RCON (round constant).The input to key schedule is the cipher key K. Key expansion generates a total of Nb(Nr + 1) words. The algorithm requires an initial set of Nb words, and each of the Nr rounds requires Nb words of key data. The resulting key schedule consists of a linear array of 4-byte words, denoted [wi], with i in the range $0 \leq i <$ Nb(Nr + 1).The subword () function takes a four byte input and applies the byte substitution operation and produces an output word. The rotword() takes a word [a0, a1, a2, a3] as input and performs a cyclic permutation to produce [a1, a2, a3, a0] as output word. The round constant word array rcon[i] is calculated using the below formula in rinjdale finite field.

rcon[i]= $x^{(i-1)}$ mod $x^8+x^4+x^3+x+1$

The first Nk words of the expanded key are filled with the cipher key. Every following word w[i] is equal to the xor of previous word w[i-1] and the word Nk positions earlier w[i-Nk]. For words in positions that are a multiple of Nk, a transformation is applied to w[i-1] prior to the XOR, followed by an XOR with a round constant Rcon[i]. This transformation consists of a cyclic shift of the bytes in a word rotword() and byte substitution subword().
But in key expansion of 256 -bit cipher if Nk=8 and i-4 is a multiple of Nk then subword () function is applied to w [i-1] prior to the xor. The algorithm for the key expansion routine is given in table 2. Thus with all the above operations the algorithm for the encryption of the data is as follows. Since it begins and ends with the add round key operation there is no wasted un keyed step in the beginning or the end. The table 3 shows the algorithm for implementation of all three AES encryption.

DOI 10.17148/IJARCCE.2015.47123

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

Algorithm for key expansion

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
word temp
i=0
while(i<Nk)
{
w[i] = word(key[4*i], key[4*i+1] key[4*i+2]i+3])
i = i+1
}
end while
I = Nk
while (i < Nb * (Nr+1)]
{
temp = w[i-1]
if (i mod Nk = 0)
temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
else if (Nk > 6 and i mod Nk = 4)
temp = SubWord(temp)
end if

w[i] = w[i-Nk] xor temp
i = i + 1
}
end while
end
```

Algorithm for encryption

```
byte state[4,Nb]
state = in
AddRoundKey(state, key Schedule[0, Nb-1])
for round = 1 step 1 to Nr–1
{
SubBytes(state)
ShiftRows(state)
MixColumns(state)
AddRoundKey(state)

key Schedule[round*Nb, (round+1)*Nb-1])
}
SubBytes(state)

ShiftRows(state)
AddRoundKey(state, key Schedule[Nr*Nb, (Nr+1)*Nb-
1])
out = state
```

2. HARDWARE IMPLEMENTATION OF ENCRYPTION ALGORITHM

Many hardware implementation of encryption algorithm using VHDL is available. In most of the case hardware implementations of AES uses only the AES-128 candidate. Some software implementation of AES192 and AES -256 are available. In the proposed architecture all candidates of AES i.e. AES-128, AES192 and AES-256 are implemented in the same device. The proposed design is implemented using VHDL coding in Xilinx ISE 9.2. Iterative looping techniques is followed to implement the entire design modules of AES encryption algorithm to the minimize hardware utilization. The key controller unit, key expansion unit, and round function unit and mix column unit everything are implemented in hardware.

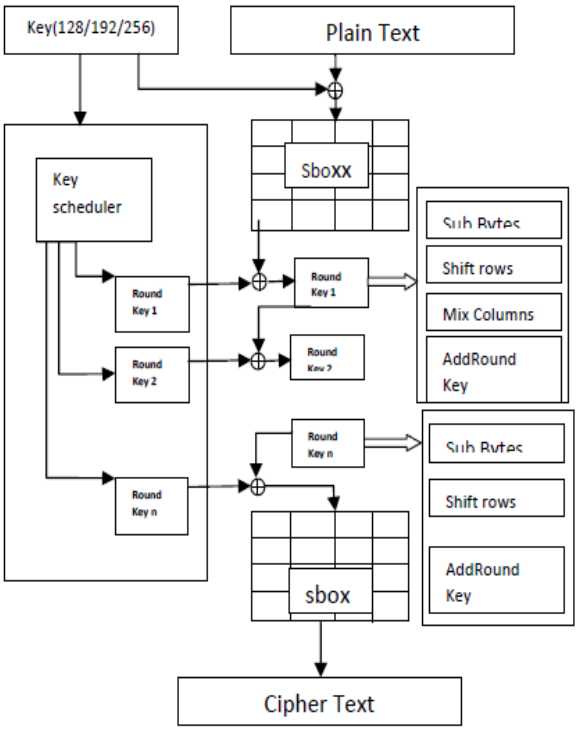

Figure 2. Flow Chart of AES Encryption Implementation

The table 2 Show s that proposed design outperforms all designs based on iterative looping in terms of area and throughput. The performance of AES-192 and AES 256 is also verified and simulation results are given. The novel architecture to implement all AES candidates in same hardware proposed is shown in figure 2. the simulation results of various AES key lengths is shown in figure 3,4 and 5.

Figure 3. Schematic Diagram AES-128/192/256 Architecture

3. CONCLUSION

The AES algorithm is an iterative private key symmetric block cipher that can process data block of 128-bits through the use of cipher keys with key length 128,192 and 256 bits. An efficient FPGA implementation of 128 bit block and keys 128, 192 and 256 bits of AES –

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

Rijindael algorithm has been presented in this paper. Optimized and synthesizable VHDL code is developed for implementation of all AES-128/192/256 bit key encryption and is verified using xilinx ISE 9.2 simulation tool. All the transformations of algorithm are simulated using an iterative design approach in order to minimize the hardware utilization. Thus it can reduce the space by enclosing three different encryption standards in a single architecture and the power consumption can also be reduced which makes it usable in battery operated network devices having Bluetooth and wireless communication devices like software radio. Throughput reaches the value of 666.7Mbps for AES-128 encryption with FPGA device XC2V6000BF957-6.

REFERENCES

[1] J. Daemen, V.Rijmen: The Rijndael Block Cipher: AES Proposal : First AES Candidate Conference (AES1) : August 20-22, 1998
[2] A. Dandalis, V.K. Prasanna, J.D.P. Rolim: A Comparative Study of Performance of AES Candidates Using FPGAs: The Third Advanced Encryption Standard (AES3) Candidate Conference, 13-14 April 2000, New York, USA.
[3] T. Ichikawa, T. Kasuya, M. Matsui: Hardware Evaluation of the AES Finalists: The Third Advanced Encryption Standard (AES3) Candidate Conference, 13-14 April 2000, New York, USA.
[4] K. Gaj, P. Chodowiec: Comparison of the Hardware Performance of the AES Candidates using Reconfigurable Hardware: The Third Advanced Encryption Standard (AES3) Candidate Conference, 13-14 April 2000, New York, USA.
[5] Xilinx VirtexTM-E 1.8V Field Programmable Gate Arrays: URL: http://www.xilinx.com: November 2000.
[6] M.McLoone, J.V. McCanny: Apparatus for Selectably Encrypting and Decrypting Data: UK Patent Application No. 0107592.8: Filed March 2001.
[7] B. Weeks, M. Bean, T. Rozylowicz, C. Ficke: Hardware Performance Simulations of Round 2 Advanced Encryption Standard Algorithms: The Third Advanced Encryption Standard (AES3) Candidate Conference, 13-14 April 2000, New York, USA
[8] Announcing the ADVANCED ENCRYPTION STANDARD (AES)" Federal Information Processing Standards Publication 197 November 26, 2001

BIOGRAPHY

B. Venkata Ramudu completed B. Tech (CSE) M. Tech (CSE) from JNTU Hyderabad. He is working as associate professor in Malla Reddy Engineering College (Autonomous) Hyderabad. His research includes data mining.

Sanjeeva Polepaka completed B. Tech (CSE) from Andhra University and M. Tech (CSE) from Acharya Nagarjuna University. He is pursuing Ph D from JNTU Hyderabad; He is working as associate professor in Malla Reddy Engineering College (Autonomous) Hyderabad. His research includes digital image processing.

DOI 10.17148/IJARCCE.2015.47123