32-Bit Advanced Encryption Standard by using Kogge-Stone Adder

V. Krupasatyā 1, G. Ravi kanth2
M.Tech Student1, Professor2
Department of Electronics & Communication Engineering
BVC College of Engineering, Rajahmundry, India

Abstract:
Now a days security is the prime part for both, the satellites communication of the electronics data and the stored data, hence encryption is important for information processing system and communication network. In this paper the designed of AES cryptography are develop by the direct method of computing the discrete linear convolution of finite length sequence. The proposed approach is easy to learn due the use of speed efficient Vedic multiplier. Since it minimize the execution time and area, so the delay and power consumption is further decrease by the compact and flexible approach in the Mix column transform which takes different approach rather than conventional multiplication previously used in AES. Model process applied in this paper is bottom-up approach. The structure style of modeling helps to easy understandable the proposed design of algorithm. AES is the symmetrical 128 bit has designed and verified in the Verilog HDL in Xilinx 14.7 tool. In this project we present a 32-bit AES by using kogge-stone adder in replace of 128-bit vedic multiplier. It performs high-performance, high throughput, and low area efficiency and low power consumption architecture for AES algorithm. The sub keys, required for each round of the Rijndael algorithm, are generated in real-time by the key- scheduler module by expanding the initial secret key, thus reducing the amount of storage for buffering. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.

I. INTRODUCTION

Since there is an evolution of wireless communication, the encrypting of data are major concern as shown in figure 1. Encryptions is the process of transfer of input text data (plain text) into the unintelligent data (cipher text) with the help of well algorithm are defined but U.S. government adopted that be used in the federal departments and agencies for protecting the important Information. According to the specifications of AES On October 2000, the NIST (national Institute of standard and technology) announced that AES encrypting algorithm as the best from other encrypting technique in the field of security, performance, efficiency, implementation capability and simplicity. Cryptography is the recognition and avoidance from the fraud and other illegal activity. The proposed AES design is the symmetric-key cryptography which involves the secret key that is only known by the user, which having the same number of bits as the palin text i.e. 128 bits. It considered that the secret key for the encryption and decryption of block of data. As for the symmetry system the secret key must be shared between the sender and the receiver for the communications purpose. The AES process is realizing in ATM, intelligence card and magnetism card.

decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. While cryptography is the science of securing data, cryptanalysis is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called attackers. Cryptology embraces both cryptography and cryptanalysis.

Problem Statement
In conventional IP forwarding, the router uses a longest-prefix match on the destination IP address to determine where to forward a packet. With MPLS, labels are attached to packets at the ingress point to an MPLS network. Within the network, the labels are used to route packets, without regard to the original packet header information. These labels can be stacked as a last in first out (LIFO) label stack, enabling MPLS flows to be combined for transport and separated later for distribution. Current proposed protocols for MPLS security, Behringer [2] and Senevirathne et al. [3] discuss two approaches to securing MPLS. Behringer [2] makes the assumption that the core MPLS network is "trusted and provided in a secure manner." We make no such assumption in our work. We assume that only the MPLS nodes themselves are secure. The physical links connecting the nodes are assumed to not be secure – we protect them using our protocol. Senevirathne et al. [3] proposes an encryption approach using a modified version of IPsec. IPsec is defined by the IETF [4], and is an all-purpose encryption protocol that includes key distribution, authentication for the IP header, and authentication and encryption for the IP payload. Senevirathne et al. [3] translate these capabilities to an MPLS environment. Their proposed system does not meet the requirements specified
above for our problem for two reasons: 1. It adds at least 128 bits to each MPLS header. This is four times the size of the MPLS header itself. This level of overhead on every packet would probably prove unacceptable to an ISP. 2. It would almost certainly add significantly more processing delay to each packet when compared to a simple encryption scheme. It does not encrypt the MPLS header (but provides authentication). Therefore the header is vulnerable to traffic analysis. We require fast and inexpensive operation, since MPLS routers are mainly routers without the full capability to do Layer 3 routing operations or are Layer 2 (ATM) switches with some additional capability. For this reason, application layer distribution designs are not applicable in our case. We also require the key exchange algorithm to be aware of the computation burden it imposes on the underlying system and communications performance. Currently, MPLS does not provide header or payload encryption. The only security function employed in MPLS is the use of MD5 [5] to sign and authenticate the control messages sent using TCP. MPLS control messages are transported using IP and do not fall under the scope of this research. They can be secured either by IPsec or any other proprietary method. Integration of the Label Distribution Protocol (LDP) security is an open issue for future study. Nevertheless, MD5 could be used for MPLS header security, since it is already present in the routers’ software. MD-5 is particularly suitable in fast re-keying and for the hash or keyed-hash functions that may need to be used.

**Advanced Encryption Standard (AES)**

The proposed algorithm of AES is based on the principle of SP-network or substitution-permutation network. [9] It is the series of a linked mathematical operations used for block cipher of AES. Such the network take the block of plain text (128 bits) and the key (128 bits) as input and applied for several alternating rounds or layers of s-box (substitution box) and p-box (permutation box) to develop the cipher text block. The rounds refers to the number of repetition of the all the layers of SP-networks. There are four Layer/ transformation of AES as defined. These are given below as

(a) Sub Byte
(b) ShiftRow
(c) MixColumn
(d) AddRoundkey

The four layer of AES is given in detail in this section. The layers have their importance in security applications. The sub byte operation is performed after around key. In the procedure of this transformation each individual byte of the state is mapped into the new state follow by the s-box table. Shift row is second step in AES. Whereas Mix column involves advance mathematical calculation. It is third step in AES. Then add Round key is last method in AES to perform encryption decryption standards for data communication.
In the SubBytes step, each byte in the array is updated using an 8-bit substitution box, the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over GF(28), 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 to avoid any fixed points (and so is a derangement), and also any opposite fixed points.

The ShiftRows step

![ShiftRows step](image)

The ShiftRows 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 ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). In the case of the 256-bit block, the first row is unchanged and the shifting for second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks.

The MixColumns step

![Mixcolumn step](image)

In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher. During this operation, each column is multiplied by the known matrix that for the 32 bit key.

\[
\begin{bmatrix}
2 & 3 & 1 & 1 \\
1 & 2 & 3 & 1 \\
1 & 1 & 2 & 3 \\
3 & 1 & 1 & 2
\end{bmatrix}
\]

The multiplication operation is defined as: multiplication by 1 means leaving unchanged, multiplication by 2 means shifting byte to the left and multiplication by 3 means shifting to the left and then performing xor with the initial unshifted value. After shifting, a conditional xor with 0x1B should be performed if the shifted value is larger than 0XFF. In more general sense, each column is treated as a polynomial over GF(28) and is then multiplied modulo x4+1 with a fixed polynomial c(x) = 0x03 · x3 + x2 + x + 0x02. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from GF(2)[x]. The MixColumns step can also be viewed as a multiplication by a particular MDS matrix in a finite field. This process is described further in the article Rijndael mix column.
In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

**Figure 5. Addroundkey step**

**DESIGN OF Kogge-Stone adders**

Stage 1 & 3 are common for both Sklansky and Kogge-Stone. Only the structure of Stage 2 gets changed. It has low depth and high node count. Minimal fan-out of 1 at each nodes to give faster performance.

An important component of digital computers is adders. Adders are used in many different parts of the digital computer. They are not only used in the Arithmetic Logic Unit (ALU) but also in address calculation. Adders are also used in multipliers and other functional units. One of the Most Prominent adders in VLSI Industry is Parallel Prefix adders1. The Parallel Prefix Adder (PPA) is one of the fastest types of adder that had been created and developed. One of the common types of parallel prefix adder is Kogge Stone adder. By using the Xilinx 14.2 software, the designs for Kogge Stone adders were developed. In this paper, Different bit of Kogge Stone adder structures can be used to execute addition and is widely used in the industry for high performance arithmetic circuits. This paper focuses on the implementation and simulation of 16-bit and 32-bit Kogge adder based on Verilog code and compared for their performance in Xilinx. Hence, this paper is significant in showing which of the adder being tested perform better in terms of computational delay based on different sizes of bits.

**Figure 6. 8-bit Kogge-Stone adder’s carry generation stage**

**Figure 7. Block diagram for final round**
The key expansion also required implementing and in some previous designs had been overlooked. One key design decision was how frequently the key must be changed and whether continued throughput is required. In this design, it was decided that throughput should be maintained during key changes and that it was desirable to change between encryption and decryption on each cycle with key changes made on similar order to the latency.

II. RESULTS

The encryptions parameter are the input plaintext, key size and output cipher text. The architecture consider the input data and key of 128 bits each and four control signal cipherkey_valid_in, clk, data_valid_in and reset. First step there be the arrangement in correct manner of 4x4 state (matrix) followed by the conversion of 16 byte from 128 bits. At beginning there is XORed operation of plaintext and input data key. Next, rest of the 9 rounds are apply with all the transformations i.e. (sub byte, Row shift, Mix column, Add round key). In the 11th round the Mix column is excluded from the loop round. The round key is generating at each rounds based on scheduling algorithm discussed earlier. The simulation of AES algorithm is done using XILINX ISE SIMULATOR of through VHDL. The plain text and the key of 32 bits will be given as the input to the design and the obtained cipher text undergoes decryption process ensure that the data generated at the end should be equal to given input. This case deals with the both encryption and decryption for first set of plain text and a key of 32 bits. The basic and common inputs for both encryption and decryption stage.

Figure 8. RTL Schematic Results of 32-bit AES by using Kogge-stone adder

Figure 9. RTL Schematic Results of 32-bit AES by using Kogge-stone adder
Figure.10. RTL schematic Results of 32-bit AES by using kogge-stone adder

Figure.11. Simulation Results of 32-bit AES by using kogge-stone adder

Figure.12. Simulation Results of 32-bit AES by using kogge-stone adder
III. COMPARISONS

Figure 13. Area utilization of 128-bit AES by using Vedic mutliplier

<table>
<thead>
<tr>
<th>Resource</th>
<th>Utilization</th>
<th>Available</th>
<th>Utilization %</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>3968</td>
<td>267600</td>
<td>1.48</td>
</tr>
<tr>
<td>LUT</td>
<td>2560</td>
<td>133800</td>
<td>1.91</td>
</tr>
<tr>
<td>I/O</td>
<td>385</td>
<td>400</td>
<td>96.25</td>
</tr>
<tr>
<td>BRAM</td>
<td>86</td>
<td>365</td>
<td>23.56</td>
</tr>
<tr>
<td>BUFG</td>
<td>1</td>
<td>32</td>
<td>3.12</td>
</tr>
</tbody>
</table>

Figure 14. Area utilization of 32-bit kogge-stone adder

<table>
<thead>
<tr>
<th>Resource</th>
<th>Utilization</th>
<th>Available</th>
<th>Utilization %</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>1536</td>
<td>267600</td>
<td>0.57</td>
</tr>
<tr>
<td>LUT</td>
<td>1040</td>
<td>133800</td>
<td>0.78</td>
</tr>
<tr>
<td>I/O</td>
<td>97</td>
<td>400</td>
<td>24.25</td>
</tr>
<tr>
<td>BUFG</td>
<td>1</td>
<td>32</td>
<td>3.12</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

The designed 32-bit AES by using kogge-stone adder IP core is analyze successfully in Verilog. The modification brought in the code are successfully tested and proved to be the message data is accurately encrypted in high security even immunity against unauthorized user. In next phase of research design the model to reduce the logic depth with add of extra memory pacifying the necessity for some looping. The Advanced Encryption Standard (AES) is the current encryption standard intended to be used by U.S. Government organizations to protect sensitive (and even secret and top secret) information. It is also becoming a (de facto) global standard for commercial software and hardware that use encryption or other security features. Rijndael is an encryption algorithm that has been designed with the state of art in the cryptographic research and is still believed very secure by most of the people. It has been designed to have very strong resistance against the classical approximation attacks, such as linear cryptanalysis, differential cryptanalysis etc. However since Rijndael is very algebraic, new algebraic attacks appeared.

FUTURE SCOPE

In the future, the AES algorithm can be implemented using 8 bits in parallel processing concept which will increase the speed and efficiency but it may require the clock frequency with more speed.

V. REFERENCES


