Analysis of New L5 Algorithm Embedded with Modified AES Algorithm in Address Allocation Schemes

Vaikkunth Mugunthan, Shrihari.S, N.Bhalaji
Department of Information Technology, SSN College of Engineering, Kalavakkam, Chennai, Tamilnadu, India.
krishna051295@gmail.com

***Abstract*—Networking is ubiquitous. In this fast-moving world, routing plays a major role for a myriad of purposes in the internet space. Address allocation is a vital component in routing. It is very much important to route the packets securely to the preferred destination. The routing of packets is not as efficient as it should have been. The current routing process is tedious and time consuming. The vulnerabilities found in the AES algorithm have been exploited by hackers and sniffers. In order to overcome these backlogs, two new mathematically validated approaches, namely, new L5 routing algorithm and modified-AES algorithm are proposed in this article. The proposed schemes circumscribe the possibilities of hacking. The mathematical exploration of the proposed schemes results in reduced time and space complexities. The newly proposed routing algorithm achieves fault tolerance, secure transmission of data and provides congestion control.**

***Index Terms*— AES (Advanced Encryption Standard); L5 Routing Algorithm; Linear Search; Network; Encryption; XOR; msb (Most Significant Bit), IP (Internet Protocol).**

I. INTRODUCTION

The protection offered to an information system in order to attain the objectives of preserving confidentiality, availability, integrity of system resources (hardware, software, information and data) is termed as Computer Security. Of late, Computer Systems are becoming more vulnerable and are prone to certain attacks. Security attacks are classified as either passive attacks or active attacks. Passive attacks include unauthorized reading of messages from a file and traffic analysis. Active attacks are those in which modification of messages or files take place. They also include the DOS and DDOS attacks. A security mechanism is any process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack. Examples of such mechanisms are encryption algorithms, digital signatures, and authentication protocols. Security mechanisms include authentication, access control, data confidentiality, data integrity and nonrepudiation. At present, however, the solutions for security problems lag well behind other areas of communications technology. Contemporary cryptography is unable to meet the requirements, in that its use would impose such severe inconveniences on the system users, as to eliminate many of the benefits [1]. Security cases are not a panacea but an orthogonal approach that directs our attention to issues that tend to slip through the cracks. Developing security cases for security-critical systems is a low risk, high-payoff path to follow [2].

A. Challenges in Cryptography

The best known cryptographic problem is that of privacy - Preventing the unauthorized extraction of information from communications over an insecure channel [1]. Secure Platform problem is one of the major issues of cryptography where malicious software may attack and modify a remote system. Repudiation comes next where ensuring comprehensive network security visibility is no easy task and Digital Signatures are not enough to achieve Non-Repudiation, i.e., the assurance that someone cannot deny something, such as the receipt of a message or the authenticity of a statement or a contract. For symmetric encryption schemes, these properties are intended to capture the protection of the confidentiality or the integrity of the encrypted messages [3]. The above mentioned challenge is dealt in this paper, by presenting a Modified AES algorithm which ensures more security while encrypting the data. The number of key sizes per block is considerably increased giving it an edge over the traditional AES algorithm.

B. Challenges in Routing

Lack of multiplexing gain, and scarcity of wavelength resources, which do not mesh well with Internet traffic that has many small and diverse flows emphasizes the importance of resource sharing [4]. Connection resiliency defines the number of node failures and disconnection events that can be sustained while preserving a given number of nodes connected to the sink [5]. Data delivery resiliency is the same as connection resiliency. However, it preserves the given number of correct packets rather than preserving the correct number of nodes. A new L5 Routing Algorithm is proposed in order to overcome the defects mentioned above and also to achieve increased efficiency. This algorithm deals with the routing of IP addresses to Local addresses and determines each address uniquely with less overhead and in minimum time period. A self-contained problem of considerable practical and theoretical interest that is related to the forwarding speed of IP traffic is the longest-prefix lookup operation, which is perceived as one of the decisive bottlenecks in IP forwarding [6]. In this paper, the presented algorithm carries out searching of addresses in a very simple and efficient manner. To keep out unwanted IP traffic, an IP firewall filters the traffic based on packet characteristics. Firewalls usually process entire IP packets [6].

II. Proposed L5 Algorithm

The L5 Routing algorithm has been developed to minimize the problems in routing the packets. The primary goal of this algorithm is to provide faster allocation of addresses. The allocation/searching of addresses are done from layer 1 and can be extended till layer 5. Hence, the name L5 Routing Algorithm.

Algorithm:

1. Start with 00 in level 0 (root level).
2. Replace any one bit from the previous bits to get the next level bits (level 1);(i.e.,)
   Level 1 node is either [01 or 10].
3. In level 2, the first child node is derived by left-padding the bits of the parent node with the msb of the parent node.
   The other child node for the particular parent will be computed based on XOR-ing the values of the parent with the first child node (obtained from the above procedure).
   Stop padding if the number of digits is greater than 4. For levels less than 4, the ip address can be obtained by left-padding the bits with zero.
4. On reaching level 3, there will be duplicates and the branches will be merged together pointing to the same node if they both yield a final value which are similar.
5. In level 4, recursive XOR-ing between the siblings in the leaf nodes (level 3) takes place and finally a set of nodes having unique addresses are obtained. If address generation cannot be done in level 4, the same can be done in level 5 by combining the two 4 bit sequences.
6. If the result equals the desired node address, it remains unchanged.
   Else, the sub sequence is to be selected, traced back to the parent and the digits replaced as necessary.

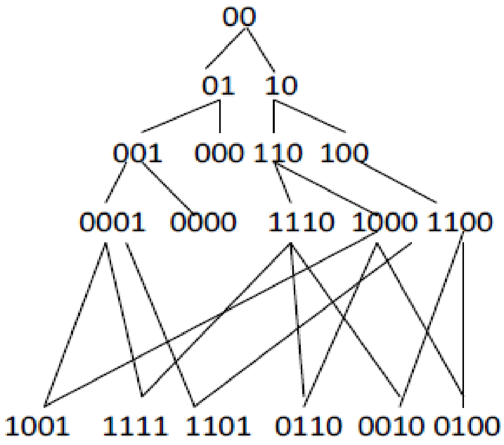

Figure 1: The L5 tree is diagrammatically depicted and the address allocation can be implemented with the help of the L5 Routing approach

Note: 1000 from 100 is omitted because of redundancy. The child nodes of 0000 are omitted because the result's value is immaterial.

Example: Considering the 122 network-
1)122.0.0.0 -root stage (00)
2)122.0.0.1 -taking the left node (01)
3)122.0.1.0 -proceed till 01 combination gets exhausted
4)122.1.0.0 -proceed till 01 combination gets exhausted
5)122.0.1.1 -proceed till 01 combination gets exhausted
Continue the procedure from 122.1.1.0 to 122.255.255.255

A. *Results and Discussion*

1. The time for searching and assigning Dynamic IP gets reduced.
2. The time complexity for linear search is O (n), whereas the proposed method has a better time complexity of O (log2h).
3. The total space is considerably reduced as there is no need to manipulate the individual numbers. Hence, the Space complexity is reduced.
4. Binary slots are found from the binary tree structure (00|01|0100…) and are converted to their corresponding numerical values.
5. Assignment of numerical values is a time consuming process, (i.e.) [x.y.z.w ->x, y, z, w] and is not efficient.
6. The proposed method uses the tree structure to find x, y, z, w in lesser time than the normal.

III. Literature Review

There are currently no algorithms which are efficient in address allocation along with secure data transmission. We have presented an algorithm that incorporates efficient routing along with strong encryption. The L5 algorithm integrated with the Modified AES algorithm overcomes the challenges discussed earlier in this article.

A. *Traditional AES Approach*

Advanced Encryption Standard (AES) algorithm is a technique used for encryption of data. AES has a fixed block size of 128 bits and a key size of 128,192,256 bits. It uses intermediate state row-column (4*4) matrices to compute the encrypted data. It uses 10,12,14 rounds for 128,192,256 key size bits respectively where the first N-1 rounds are same, whereas the final Nth round is different for each key size. The data gets encrypted in four stages and decryption is also possible in similar manner. Keys' expansion is the process where round keys are derived from the cipher keys.

Initial Round:

Step 1: AddRoundKey- bitwise XOR is used to combine each byte of the state to the block of the round key.

Step 2: Rounds- it has 4 rounds for each cycle.

1. Sub Bytes- each byte is replaced with another one derived from a lookup table.
2. Shift Rows- The first row is always unchanged whereas the remaining three rows of the state are shifted a certain number of steps.
3. Mix Columns- Each particular column is being

transformed to a different one using XOR operations.

Step 3: AddRoundKey (Step 1) is done again at the end of each round.

Final Round - (no Mix Columns): SubBytes, Shift Rows, AddRoundKey are the default operations in this round. The major factors for a quick acceptance for Rjindael is the fact that it is available royalty-free, and that it can be implemented easily on a wide range of platforms without reducing the bandwidth in a significant way [7]

B. Limitations of Traditional AES Approach

One of the major limitations of traditional AES algorithm is the fixed key size, (i.e.) it supports only one of these bit sizes [128,192,256 bits]. It requires multiple rounds of communication to encrypt the data precisely. It has a simple algebraic structure and hence is a challenge to implement in GCM mode in terms of security and as a permanent methodology. AES was introduced to replace the traditional DES. Brute force attack is the only effective attack known against this algorithm [8]. AES showed poor performance results compared to other algorithms since it requires more processing power [8]. AES decryption algorithm nevertheless allows InvMixColumn and AddRoundKey to be reordered if an additional InvMixColumn operation is performed on most of the Round Keys (except the first and the last Round Keys)[9].

C. AES and its Applications

AES is widely implemented and its use is aggrandizing day by day. Different pipelined implementations of the AES algorithm as well as the design decisions and area optimizations that lead to a low area and high throughput AES encryption processor are presented by Hodjat, Alireza, and Ingrid Verbauwhede [10]. Furthermore, composite field arithmetic is employed to reduce the area requirements, and different implementations for the inversion in subfield (24) are compared and this is presented by Zhang, Xinmiao, and Keshab K. Parhi [11]. A compact FPGA architecture for the AES algorithm with 128-bit key targeted for low-cost embedded applications is presented. However encryption, decryption and key schedule are all implemented using small resources of only 222 Slices and 3 Block RAMs. This implementation easily fits in a low-cost Xilinx Spartan II XC2S30 FPGA and can encrypt and decrypt data streams of 150 Mbps, which satisfies the needs of most embedded applications, including wireless communication and is done by Chodowiec, Paweł, and Kris Gaj [12]. On the whole, improvement is done in the performance of AES by increasing its key block size so that it can be easily implemented for more number of bits. The reduction of overhead and time complexity associated with the AES Algorithm is accomplished by using the 24 bit padding algorithm, as shared in this article, along with the modified AES algorithm.

D. Modified AES Approach

Algorithm:

1. Add round key to the given input data.
2. Shift rows: 1st row doesn’t change, 2nd row is shifted right by 1 place, 3rd row is shifted right by 2 places, and 4th row is shifted right by 3 places to obtain the state matrix.
3. Mix columns: Use the default derived matrix for computing this step.

Default matrix is:

2 3 1 1
1 2 3 1
1 1 2 3
3 1 1 2

In mix columns,

1. When multiplying an element in the matrix by 1, it remains unchanged; the element is converted into its binary form.
2. When multiplying an element in the matrix by 2, the binary form of the element is shifted left by 1 bit and 0 is padded at the end to complete the 8 bits; the result is XNOR-ed against 11100100(E4) to get the final value.
3. When multiplying an element in the matrix by 3, 3(11) is split as 10 XOR 01. Element * 01 (i.e., 1 in decimal) = Element. But, for Element * 10 (i.e., 2 in decimal), the same procedure mentioned in point 2 is to be followed. Finally, the values obtained from Element*01 and Element*10 are to be XOR-ed.

   The final value to be entered will be the value computed by this formula:

   02*A1 XOR 03*A2 XOR 01*A3 XOR 01*A4 – First Element

   01*A1 XOR 02*A2 XOR 03*A3 XOR 01*A4 – Second Element

   01*A1 XOR 01*A2 XOR 02*A3 XOR 03*A4 – Third Element

   03*A1 XOR 01*A2 XOR 01*A3 XOR 02*A4 – Fourth Element

   where A1, A2, A3, A4 are the elements of the input [4*1] matrix respectively.
4. Add a round key.
5. Repeat the steps for N Rounds.

E. Mathematical Proof and Discussions

This methodology is applicable for key sizes of 128,192,256 bits and for all regular rounds in the loop. (i.e.) N-1 rounds where N=10, 12, 14 for key sizes of 128,192,256 bits respectively.

Round 1:

Step 4: Add round key to the given input data Subkey [0].

Step 5: Shift rows: The 1st row doesn’t change, the 2nd row is shifted right by 1 place, the 3rd row is shifted right by 2 places, and the 4th row is shifted right by 3 places to obtain the state matrix.

Step 6: Suppose the given input is as follows:

E2 02 03 01 01
31 01 02 03 01
4A 01 01 02 03
26 03 01 01 02

Methodology followed here is Mix-Columns obtained by Re-arranging the matrices.

02 03 01 01 E2
01 02 03 01 31

01 01 02 03 4A
03 01 01 02 26
The first element is computed as:
02*E2 XOR 03*31 XOR 01*4A XOR 01*26
Similarly the remaining elements are computed from the default matrix using the above methodology to fill the State matrix of Mix-Columns step.

Step 7: Add a round key [Sub key (N)] and perform encryption

Step 8: Repeat the steps for N Rounds

IV. PADDING

Along with the modified AES algorithm, padding is carried out to support appending extra bits. 24 bits are left-padded to their respective key sizes [128,192,256] making them support [152,216,280] bits respectively.

Algorithm:

1. Get 2 numbers (J and I), where J is greater than or equal to I.
2. XOR the two numbers.
3. Perform AND operation for the result obtained in step (2) with I and store the result as K.
4. If the obtained value is less than or equal to J and greater than or equal to I, left-pad the keys with K
5. If the condition mentioned in step 4 is not satisfied, repeat from step 1 till the condition is satisfied.
6. Repeat till 24 bit padding is obtained.

Discussions:

1. It reduces the total number of XORs to almost half.
2. The number of logical gates can be reduced dramatically. The number of operations to be performed is lesser.
3. Less Overhead, lower space consumption and faster execution are some of the notable advantages.

When using a XNOR gate, an Inverter along with the XOR gate is no longer necessary. The XNOR gate has the added advantage of requiring little space, less material and is less complicated as compared to the combination of Inverter and XOR gate.

V. CONCLUSION

The L5 Algorithm presented in this paper provides an efficient, secured, fast and reliable Address allocation. The modified AES Algorithm proposed in this literature gives the comfort of adding more bits in a key block size and also reduces the time and space complexities. The time complexity for the routing process is reduced from O (n) to O (log2h). Thus, a fault-tolerant, mathematically validated and a highly reliable algorithm is proposed for secure transmission of data in IPv4 networks. In future, we intend to simulate the proposed algorithms over the appropriate test bed for further validation and also to implement the same in an IPv6 based protocol.

REFERENCES

[1] Diffie, Whitfield, and Martin E. Hellman. "New directions in cryptography."*Information Theory, IEEE Transactions on* 22, no. 6 (1976): 644-654.

[2] Knight, John. "The Importance of Security Cases: Proof Is Good, But Not Enough." *Security & Privacy, IEEE* 13, no. 4 (2015): 73-75.

[3] Maurer, Ueli, Andreas Rüedlinger, and BjörnTackmann. "Confidentiality and integrity: A constructive perspective." In *Theory of Cryptography*, pp. 209-229. Springer Berlin Heidelberg, 2012.

[4] Huang, Hong, and John Copeland. "Optical networks with hybrid routing."*Selected Areas in Communications, IEEE Journal on* 21, no. 7 (2003): 1063-1070.

[5] Testa, Alessandro, et al. "Heuristic strategies for assessing wireless sensor network resiliency: an event-based formal approach." *Journal of Heuristics*21.2 (2015): 145-175.

[6] Chao, H. Jonathan, Mikael Degermark, and Nick McKeown. "Next-generation IP switches and routers." *Selected Areas in Communications, IEEE Journal on*17, no. 6 (1999): 1009-1012.

[7] Daemen, Joan, and Vincent Rijmen. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.

[8] Thakur, Jawahar, and Nagesh Kumar. "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis."*International journal of emerging technology and advanced engineering* 1, no. 2 (2011): 6-12.

[9] Rouvroy, Gaël, François-Xavier Standaert, Jean-Jacques Quisquater, and Jean-Didier Legat. "Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well suited for small embedded applications." In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, vol. 2, pp. 583-587. IEEE, 2004.

[10] Hodjat, Alireza, and Ingrid Verbauwhede. "Area-throughput trade-offs for fully pipelined 30 to 70 Gbits/s AES processors." *Computers, IEEE Transactions on*55, no. 4 (2006): 366-372.

[11] Zhang, Xinmiao, and Keshab K. Parhi. "High-speed VLSI architectures for the AES algorithm." *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 12, no. 9 (2004): 957-967.

[12] Chodowiec, Paweł, and Kris Gaj. "Very compact FPGA implementation of the AES algorithm." In *Cryptographic Hardware and Embedded Systems-CHES 2003*, pp. 319-333. Springer Berlin Heidelberg, 2003.