HNS Advanced Encryption Standard: An Enhanced Security Approach for IoT Communication

Mohamed Kassab¹*, V Nithya¹, and Mohamad Sadek Mokyed²
¹Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamilnadu, India
²Department of Computer Engineering, University of Debrecen Official, Hungary
Email: *mk7243@srmist.edu.in

Abstract. AES is considered as synchronize encryption whose keys are the same in the sender and the receiver. All the researches work to enhance the security in this type of encryption by making the keys nonlinear. If attackers know any one of the sub-keys, this will lead to knowing other keys in a not difficult way. To make it more nonlinear and more difficult to be attacked, it should be more balanced between some ones and zeros in the ciphertext. We make some changes in the key expansion function to get results more balanced than the original design AES.

Keywords: AES; Balance; IoT Security; Key dependent round

1. Introduction

There are two types of encryption: in the first type, called synchronize encryption, the sender uses the same key that the receiver uses. In Asymmetric encryption, private key and public key are used from the sender and the receiver[1]. AES is symmetric key encryption. In some applications related to security, the AES algorithm is used like ZigBee, Internet of things (IoT), and other applications. AES's structure is considered strong, but it can be enhanced to make it more nonlinear to prevent knowing all sub-keys if any one of them is known from attackers. The balance parameter depends on the number of ones and zeros in ciphertext [2]. The best percentage is 50% which means 64 ones and 64 zeros in the ciphertext, and it is the most nonlinear percentage. The proposed model for the enhanced Algorithm is HNS AES, and it represents the first letters of the author's names.

2. Structure of AES

AES is a synchronized encryption algorithm with a strong structure compared to other algorithms like DES because the structure has ten rounds related to sub-keys. Each round has four functions: add round key function, sub byte function, shift row function, and mix column function. Mix column is not found in the tenth round. Figure 1 shows the AES procedure.
Add round key function is XOR between the two inputs main key and plain text. The sub-byte function substitutes the input, which is 128 bits by another bit. It replaces those 8 bits with 8 bits. Figure 2 shows the Sub byte function.

**Figure 1:** AES procedure.
Shift row function: it rotates each row of the input with different numbers of shifting. Mix column: it multiplies the input with a constant state. Figure 3 shows the Mix column function.

3. Related Work

Lots of previous studies have attempted to enhance the AES algorithm’s efficiency. By having its phases mainly dependent, researchers have mainly focused on creating a stronger AES. The second stage is Sub Bytes in the encryption process. This stage employs an S-box as a search [3]. The fixed of S-Box is defined as a matrix of 16 x 16 composed of values of bytes. Byte values are the changing of all possible values of 256 or 8 bits. Accessing of Sbox doesn't have to have a key. For Sbox, the studies tried to make it able to reach all beneficiaries just with a key. That insure improving safety.

Get better results according to avalanche effect parameter and ensure encryption provident by adding xor function to algorithm structure. [4-5] by the key, the S-Box has been turned to be dynamic, which he used to be static in the original AES, so the complexity is increased. [6] This study
introduces a system with specifications. Composing enhanced safety. So that the velocity has speeded up and consumption of energy has been reduced.

This cryptosystem's possibility depends on the XOR of one picture from a picture information base with the unmistakable discourse beforehand encryption[7]. It also implanted the record number of this picture in the information[8] base as a most un-huge digit watermark into discourse [9].

Discusses the utilize of reversible logic to moderate the energy in regular AES as reversible entryways offer in a perfect world zero interior force scattering[10]. The proposed work is designed and implemented using the Xilinx tool and verified on Virtex-5 FPGA using chips. Results are discussed w.r.t power, the needed hardware and the delay using Feynman's gates, and costs of quantum. In this research, the author indicates that by collective the number of keys, the time of implementation has been increased as well. Each key that is added to the rotation of the Algorithm is increased

4. Proposed Model
The key expansion function in the original AES algorithm is responsible for generating 11 keys, where the input is the main key. Each key consists of 128 bytes or four words, and each round of key expansion function generates a new key.

The main key consists of w0, w1, w2, w3. And the first key consists of w4, w5, w6, w7, and we stay in this way till reaching the tenth key, which consists of w40, w41, w42, w43.

The first key is related to the main key, and the second key is related to the first key. The tenth key related to the ninth key in Figure 4 shows the original design's KEY expansion structure.

![AES Key Expansion](image)

(a) Overall algorithm

(b) Function g

Figure 4: KEY expansion structure in the original design.
Generating the words which are not divided by four is easier than words accept dividing by four because there is a g function.

a) The rotate word does 8 bits, shifting toward the left on the input. For example, if the input is [A0 A1 A2 A3], the output will be [A1 A2 A3 A0].

b) Substitute Word plays 8 bits' replacement on every 8 bits of its input word, by S-box.

c) With around constant, Rcon [j], the outcome of stages 1 and 2 is XORed [j] [11]. The round constant is a term where the 3 bytes on the right are continuously 0. Thus, with Rcon, the result of an XOR of a term is to do only an XOR on the word's left-most byte. For every phase, the round constant is changed and is defined as Rcon [j] = (RC[j], 0, 0, 0), RC [1] = 1, RC[j] = 2 RC [j -1] in addition multiplication over the GF (28) sector [12]. The hexadecimal value of RC[j] is Table RC array values.

| j  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|----|----|----|----|----|----|----|----|----|----|----|
| RC[j]| 01 | 02 | 04 | 08 | 10 | 20 | 40 | 80 | 1B | 36 |

Besides, the key expansion function in the HNS AES design will be as in Figure 5.

**Figure 5:** Key expansion structure in HNS AES design.

The proposed key expansion in the HNS AES algorithm exchange zeros in g function by parts of the main key as shown in next Figure 6:
The RC[j] will be as original
And the rest instead of zeros, we will put as in table where
w1 is the first two hex digits of the main key.
w2 is the second two hex digits of the main key.
W3 is the third two hex digits of the main key and so on.

Figure 6: enhance RC values.

Figure 6 shows enhance RC values. These changes lead to the best results according to the balance parameter.

5. Performance Analysis: Non-Linearity
- Balance parameter:
The key expansion function generates ten keys from the main key, so making it non-linearity leads to
if anyone knows a key[12], it will be difficult to derive other keys. Balance parameter tests the non-linearity by counting ones in the ciphertext[13]. The best result is 64 ones and 64 zeros of 128 bits' text[14]. The next steps show the results when the key is constant.
Key = \text{fe78ff119f83cca91298563e9fec69ff}
and the plain texts are changed (in Table 1).

| ID  | The plain texts                  | N. 1's in original AES | B%   | N. 1's in HNS | B%   |
|-----|---------------------------------|------------------------|------|---------------|------|
| 1   | cc19ffe0987ffec1987ffec8764d12ff | 68                     | 53.12| 61            | 47.65|
| 2   | cc19ffe1987ffec1987ffec8764d12ff | 75                     | 58.59| 72            | 56.25|
| 3   | cccffff1987ffec1987ffec8764d12ff | 59                     | 46.09| 59            | 46.09|
| 4   | ffffffff1987ffec1987ffec8764d12ff | 59                     | 46.09| 62            | 48.43|
| 5   | fffffff1111ffec1987ffec8764d12ff | 74                     | 57.81| 58            | 45.31|
| 6   | fffffff1111ffec1117ffec8764d12ff | 57                     | 44.53| 60            | 46.89|
| 7   | fffffff1117ffec1117ff118764d12ff | 54                     | 42.18| 64            | 50   |
| 8   | fffffff1117ffec1117ff118444d12ff | 53                     | 41.40| 66            | 46.87|
| 9   | fffffff1117ffec1117ff118444ff2ff | 62                     | 48.43| 64            | 50   |
| 10  | fffffff1117ffec1117ff118444ffffff | 71                     | 55.46| 64            | 50   |
| 11  | fffffff1117ffec1117ff118444ffffff | 71                     | 55.46| 69            | 53.60|
| 12  | fffffff1117ff11ff118444ffffff   | 71                     | 55.46| 67            | 52.34|
| 13  | fffffff1117ff11ff118444ffffff   | 70                     | 55.46| 62            | 48.43|
The next steps show the results when the plain text is constant ffffffff1111711fff11fff118444fffffff
And the key is changed[15]. Figure 7 shows the comparison based on balance measures for fixed keys. Figure 8 shows a comparison based on balance measure for fixed plain text (Table 2).

Table 2: Fixed plain text

| ID | The key                              | N. 1’s in original AES | B%  | N. 1’s in HNS | B%  |
|----|--------------------------------------|------------------------|-----|---------------|-----|
| 1  | cc19ffe0987ffec1987ffec8764d12ff      | 57                     | 44.53 | 69            | 53.90 |
| 2  | cc19fff1987ffec1987ffec8764d12ff      | 69                     | 53.90 | 59            | 46.09 |
| 3  | ccffff1987ffec1987ffec8764d12ff       | 56                     | 43.75 | 72            | 56.25 |
| 4  | ffffffff1987ffec1987ffec8764d12ff     | 71                     | 55.46 | 65            | 50.78 |
| 5  | ffffffff11117ffec1987ffec8764d12ff    | 58                     | 45.31 | 60            | 46.87 |
| 6  | ffffffff11117ffec11117ffec8764d12ff   | 77                     | 60.15 | 60            | 46.87 |
| 7  | ffffffff11117ffec11117ffec118764d12ff | 71                     | 55.46 | 71            | 55.46 |
| AVG|                                      | 65.5714                | 51.22 | 65.1429       | 50.88 |

Figure 7: Comparison based on balance measure for fixed keys.

Figure 8. Comparison based on balance measure for fixed plain text.

6. Conclusion
We enhance the AES algorithm by making it more nonlinear by replacing zeros in the g function in the key expansion function with different parts of the main key. We get balancer results compared to the original AES. This lead to making it more immune again analysis attacks.

In our examples, when the key is constant, the balance AVG in HNS is 49.37 comparing to 50.72 in the original AES. This mean 0.09% is achieved in the balance parameter. Besides, when the plaintext
is constant, and the keys are varied, the balance AVG in HNS is 50.88 comparing to 51.22 in original AES. This means HNS is better than AES by 0.34%, according to the balance parameter.

References
[1] Awll, A. H., & Aziz, S. M. (2021). Secure Device to Device Communication for 5G Network Based on improved AES. The Scientific Journal of Cihan University – Sulaimaniya, 5(1), 57-67.
[2] Karaki, B. J., & Mahmoud, M. S. (2021). Quantised scaled consensus of linear multiagent systems on faulty networks. International Journal of Systems Science, 1-15.
[3] Rohini Hongal, Jyoti H, Rajashekar S "An Approach towards Design of N-Bit AES to Enhance Security using Reversible Logic" Communications on Applied Electronics (CAE) – ISSN: 2394-4714 Foundation of Computer Science FCS, New York, USA Volume 7– No. 22, November.
[4] B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.
[5] Ajeet Singh, "A New Approach to Enhance Avalanche Effect in Aes to Improve Computer Security" Inform Tech Softw Eng 2015, 5:10.4172/2165-7866.1000143
[6] Hasan, H. A. A., Mohammed, S. M., & Ameer, N. H. A. (2021). ADVANCED ENCRYPTION STANDARD USING FPGA OVER NETWORK. EUREKA: Physics and Engineering, (1). 32-39.
[7] Alanezi, A., Abd-El-Atty, B., Kolivand, H., El-Latif, A., Ahmed, A., El-Rahiem, A., ... & S Khalifa, H. (2021). Securing Digital Images through Simple Permutation-Substitution Mechanism in Cloud-Based Smart City Environment. Security and Communication Networks, 2021.
[8] Emad Mossa, "Security enhancement for AES encrypted speech", Int J Speech Technol (2017) 20:163–169 DOI 10.1007/s10772-017-9395-3 in communications
[9] Alanezi, A., Abd-El-Atty, B., Kolivand, H., El-Latif, A., Ahmed, A., El-Rahiem, A., ... & S Khalifa, H. (2021). Securing Digital Images through Simple Permutation-Substitution Mechanism in Cloud-Based Smart City Environment. Security and Communication Networks, 2021.
[10] Amudha, S., Snehalatha, N., & ShinyAngel, T. S. (2016). SMS Controlled Smart Home System in IoT. International Journal of MC Square Scientific Research (IJMSR), 8(1), 1-8.
[11] Bothe, A., Bauer, J., & Aschenbruck, N. (2019, September). RFID-assisted Continuous user authentication for IoT-based smart farming. In 2019 IEEE International Conference on RFID Technology and Applications (RFID-TA) (pp. 505-510). IEEE.
[12] Radanliev, P., De Roure, D. C., Maple, C., Nurse, J. R., Nicolescu, R., & Ani, U. (2019). Cyber Risk in IoT Systems.
[13] Ali, A., Ahmed, M., Imran, M., & Khattak, H. A. (2020). Security and privacy issues in fog computing. Fog Computing: Theory and Practice, 105-137.
[14] Gu, T., Abhishek, A., Fu, H., Zhang, H., Basu, D., & Mohapatra, P. (2020, August). Towards Learning-automation IoT Attack Detection through Reinforcement Learning. In 2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks" (WoWMoM) (pp. 88-97). IEEE.
[15] Akanksha, E., Debnath, A., & Dey, B. (2021, January). Extensive Review of Cloud-Based Internet of Things Architecture and Current Trends. In 2021 6th International Conference on Inventive Computation Technologies (ICICT) (pp. 1-9). IEEE.