ARMSTRONG NUMBER ENCRYPTION STANDARD FOR SMART DEVICES - AN IOT BASED ENCRYPTION ALGORITHM

Polasi Sushma, Assistant Professor, Loyola Academy Degree and PG College, Secunderabad & Research Scholar, Department of Computer Science, Osmania University, Hyderabad,Telangana, India :
Prof. V. V. Hara Gopal Department of Statistics, Osmania University, Hyderabad, Telangana, India :
polasi.sushma@gmail.com; haragopal_vajjha@yahoo.com

Abstract: In the ERA of internet of things, where smart living has took the front seat, Smart devices enhanced the quality of lifestyle and improved the heterogonous connectivity. These are everyday things with intelligence and unconventional computing capabilities involving machine learning and artificial intelligence. In simple terms, smart devices in different sector like smart homes, smart buildings, and smart cities are where things or devices can communicate with each other and also can be operated with a smart phone applications from a remote location. These are the endpoints of networks with high independent workloads. Smart devices are now ranging from small home things to a huge manufacturing industry things. These devices can send and receive messages over the internet and are in a open threat of being compromised. IoT provides many protocols for the devices to communicate with each other. Few of the protocols popularly used for the communications are MQTT, CoAP, XMPP, AMQP. These protocols are generally light in weight as they work on IoT. The major criteria of concern in these protocols is the light weight encryption and decryption algorithms for the secure communication between the devices. Smart devices have constrained memory and require less power consumption for computations. Complex encryption algorithm with huge computations and memory space consume more power and storage space making the smart device slower and drain quickly. This paper focuses at providing security using Armstrong number encryption standard algorithm to work on the binary data. It enhances the security of smart devices by providing confidentiality and integrity of the data with simple computations.

Keywords: Encryption, Armstrong Number encryption standard, security, confidentiality

INTRODUCTION

Internet of Things is a new revolution of the Internet that is budding exponentially and congregation momentum rapidly, driven by the enhancements in senor networks, mobile devices, wireless communication networking and cloud technologies. Experts forecasted that eighty billions devices or things will connect to the Internet by 2025[3]. Simply, Internet of Things can be defined as the inter connection of physical and virtual objects that have unique identity facilitated by intelligent applications that make energy, retail, logistics, agriculture, industrial control, and many other domains [6]. These things communicate with each other by sending and receiving messages. There are many threats while the data is in transit like man in middle attack, replay attack, distributed denial of service, brute force attack etc. [4]. There is a need for protecting this data by the security managers and the IoT users [5]. One of the best protection measure is to encrypt the data. There are many encryption algorithms in use like DES, AES, RSA, Blowfish etc. [8]. These algorithms are proved to be efficient on the public networks, but for the IoT constraint devices with less power and storage space computationally they consume more power [12]. We require algorithms which are computationally simple, light weight, yet provide a high level of security for the data communications between the IoT devices [9]. This paper focuses on protecting the data by using Armstrong numbers encryption standard algorithm.

CRYPTOGRAPHY

The processes of protecting the sensitive business and user data from unauthorized access over the network is called as Cryptography [1]. There are two types of commonly used cryptography

1. Symmetric/secret key cryptography
2. Asymmetric/public and private key cryptography

Only one secret key is used both by the sender and the receiver for the encryption and decryption in the symmetric key encryption. This provides confidential communications between the sender and the receiver [2]. Asymmetric key encryption uses two keys: public key for encryption and private key for decryption. This provides confidential and authenticated communication between the sender and the receiver.

Based on the requirement of the light weight encryption algorithm for the smart devices, this paper uses symmetric key encryption called Armstrong number encryption standard for securing the data communications. Client authentication can be done by using usernames and passwords and data confidentiality can be ensured by secret key encryption. It is always advised to use different key for each session to secure data from being compromised. This algorithm is designed to work on the binary data. It enhances the security of business and user data by providing confidentiality and integrity of the data.

ARMSTRONG NUMBER ENCRYPTION STANDARD

Armstrong Number Encryption Standard is a process used to encrypt and decrypt plaintext inputs which is in the form of binary data by creating a cryptographic chain wherein each cipher text block is dependent on the last. Armstrong Number Encryption Standard (ANES) is a cryptographic method used for turning plaintext into cipher text and back again. ANES uses Armstrong number as a secret key and also uses a random number as an Initialization Vector to enhance the security of data. The chaining feature of ANES can identify the change of even a single bit in transit. Destination system can identify that the data is compromised and ask for retransmission of data. Same block of data though repeated, will not be same for next time because of IV initialization vector usage.

ANES basically uses simple operations like

- Circular left shift
- Circular right shift and
- XOR operations.

These simple computations makes the algorithm a light weight algorithm. Essentially, in ANES, each plaint text block is XORed (numerically combined) with the previous cipher text block and then encrypted. An XOR is a coding mechanism used to combine different inputs. It is used in this case to facilitate the combination of plaintext blocks and encryption keys. The process repeats itself until all plaintext blocks have been successfully turned into cipher textblocks.

ANES uses arbitrary size plain text. Based on the block size and the number of blocks too, the algorithm works.

For each session this value may change, keeping the sessions unpredictable and secure from the hacker.

The data published from source to destination remains encrypted by using Armstrong number encryption standard

Mathematical approach

Encryption process

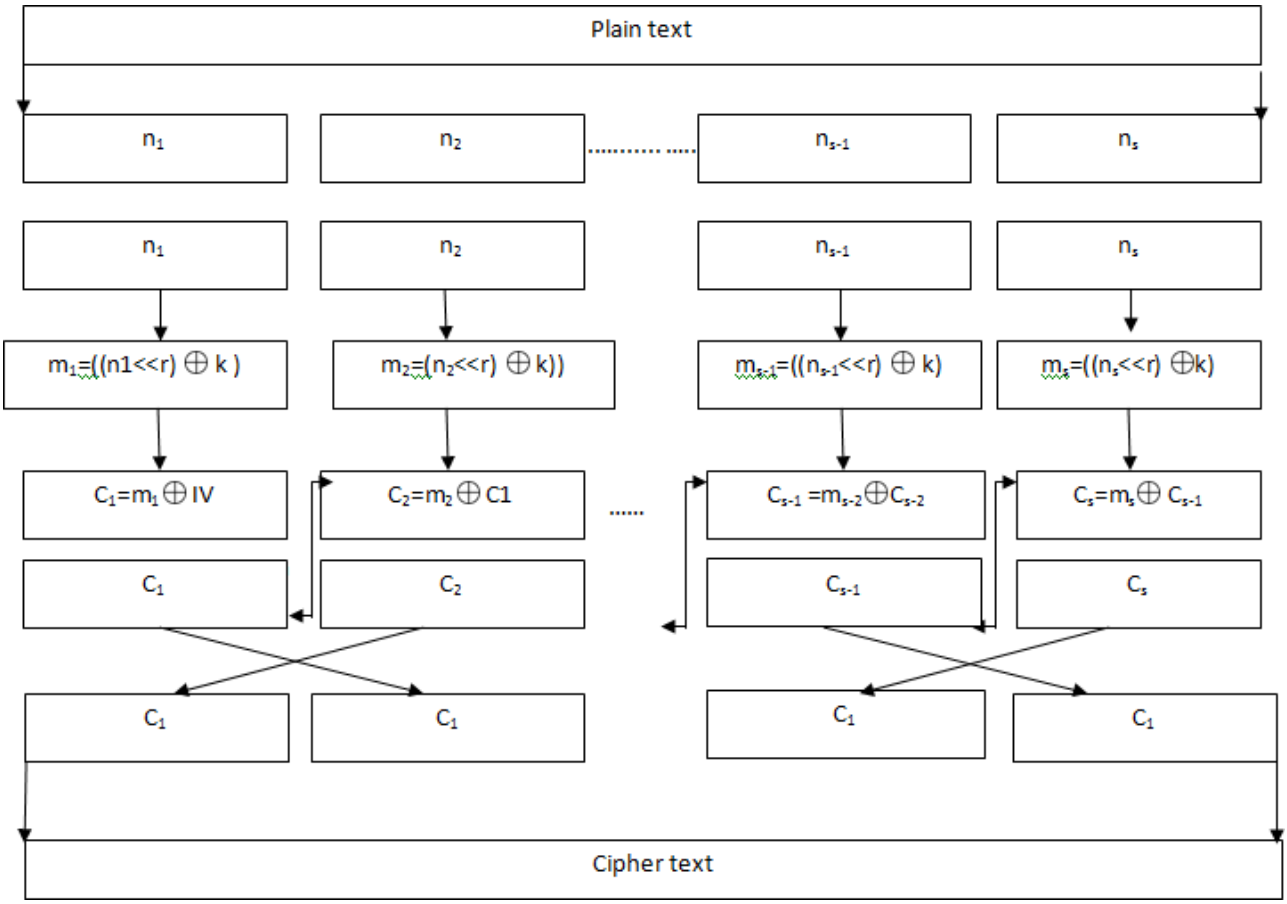

FIGURE. 1 ANES encryption

Plain text is converted to binary stream of data as depicted in Fig.1

Step 1: The binary stream of data is divided into ‘s’ consecutive ‘b’ bit blocks.Say n1, n2, n3 ns-1, ns Last block is parsed with 0 bits to form ‘b’ size bit block.

Step 2: value ‘r’ is calculated as number of bits in each block / total number of blocks

Then the first block n1 is circular left shifted by r times and then XORed with Armstrong number ‘K’ and let the resultbe ‘m1’

Step 3: m1 is XORed with a random number ‘IV’ which is also called as initialization vector to generate the first blockof cipher. Let it be C1. Initialization vector can change for every encryption on the text.

Step 4: Transformation operation is performed where first and second blocks are interchanged. Similarly, third and fourth blocks. Finally last and last before blocks are interchanged to form the final cipher text, ‘C’. If we don’t have even number of blocks then last block is forwards as it is to form the final block of cipher text.

Step 5: This C is transferred over the network to the destination node.

Decryption process

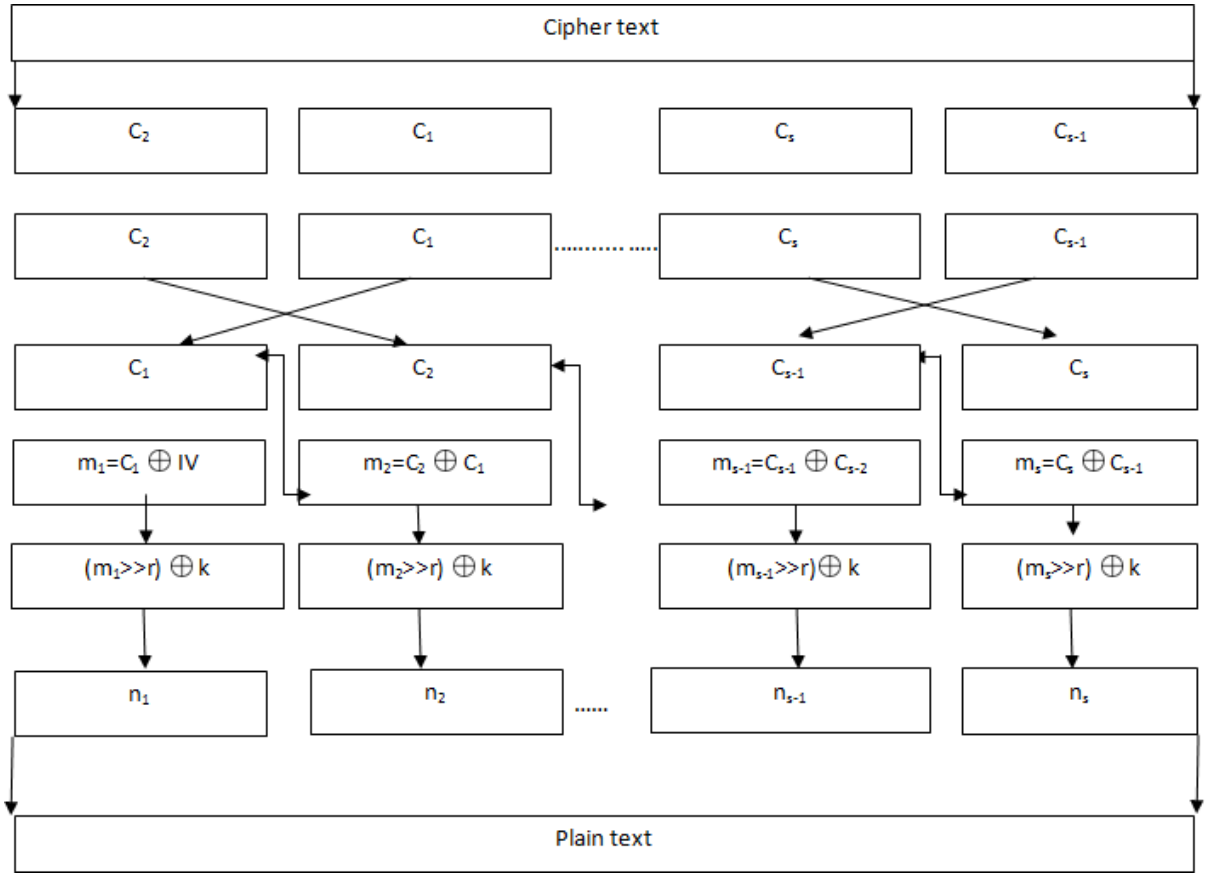

FIGURE. 2 ANES Decryption

Decryption is a process of converting cipher text to plain text.Fig. 2 depicts the steps involved in the ANES decryption.

Step 1: Cipher text "C" is divided into 's' cipher blocks of size 'b'.Say C1, C2, C3, C4...Cs-1, Cs.

Step 2: Reverse transformation is performed on these blocks, so that the cipher blocks sets in order. First block is interchanged with the second block Third with fourth and finally 's-1' block with 's' block.

Step 3: First block C1 is XORed with random number IV which is also called as initialization vector to obtain m1 block. For the next block of cipher previous block will serve as the input in the place of initialization vector. In this ways the blocks are chained.

C1-> C2, C2->C3...Cs-1 to Cs

Step 4: This 'm1' block is XORed with Armstrong number 'K' and then right shifted.This 'n1' will be the input plain text first block.

Step 5: above steps are repeated for the remaining blocks as well to obtain the plain text blocksn1, n2, n3, n4.....ns.

Experimental Investigations

ANES encryption

Consider the plain text 'Temp 26' message which comes from a thermostat. This message has to be encryptedusing ANES algorithm.

Initially the plain text has to be converted to binary format. Plain text: P:
01010100010001010100110101010000001000000110010001

101101111010100001O110
Step 1: this plain text has to be divided to 16 bit blocks. Considering the size 'z' of the block to be 16 bits. This plaintext will be divided into s=5 blocks of plain text n1, n2, n3, n4, n5
n1= 0101010001000101 n2=0100110101010000 n3=0010000000110010 n4=0011011000001010 n5=0000101000000000
Last block is parsed with 0's to form a block of size 16 bits.Consider the key which is an Armstrong number 'K'=153=0000000010011001
Consider the random number or the initialization vector'IV'=9=0000000000001001
Step 2: calculate r=z/sSo the value of r=3
Perform circular left shift 'r' times on the first block n1, then XOR it with Armstrong number 'K' (n1<<r) Fig. 3 depicts this circular left shift operation when r=3

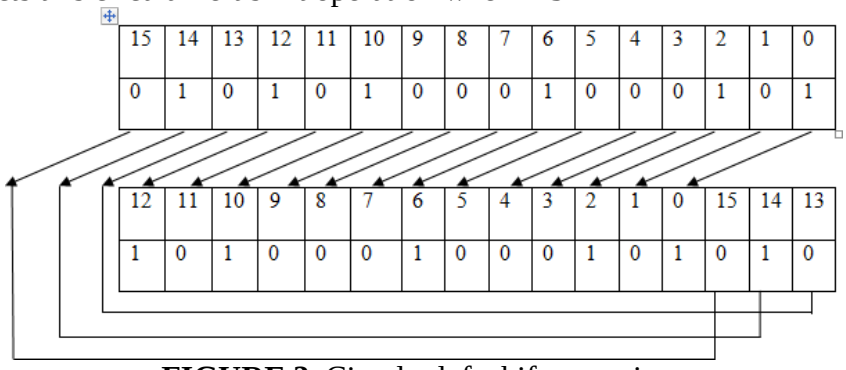

FIGURE 3. Circular left shift operations

m1= (n1<<r) ⊕K

= 1010001000101010 ⊕ 0000000010011001
= 1010001010110011
Step 3: 'm1' is XORed with Initialization vector which is a random numberm1⊕ IV=1010001010110011 ⊕ 0000000000001001 C1=1010001010110011
Similarly obtain the remaining cipher blocks C2, C3, C4, and C5.
For the second block m2, C1 will the input in step 3 instead of IV. This way the blocks are chained.Step 4: perform transformation of the cipher blocks to obtain cipher text C.
Cipher blocks C1 and C2 are interchanged, C3 and C4 are interchanged, and C5 block is forwarded as it is to form thecipher text C.
Blocks are positioned as followsC= C2, C1, C4, C3, C5
Step 5: Transmit the cipher text C on the network.Cipher text C:
1100100010100001101000101011101011001001101010011111
10010110000001010011111 1000

This way the payload of MQTT packet is encrypted in the application layer to provide the confidentiality to the text.

*ANES Decryption:*The cipher text C is divided into 5 consecutive 16 bit blocks C: 0110010001010000110100010101110101100100110101000101111100101100000000101001111110000
Step 1: Cipher text "C" is divided into 's=5' cipher blocks of size '16 bits'.C2=0110100010100001 C1=1010001010111010 C4=0111100101100001 C3=0111100101100001 C5=0000101001111110
Step 2: Perform transformation on the blocks to arrange them in order.C1, C2, C3, C4, C5
Step 3: First block C1 is XORed with random number IV which is also called as initialization vector to obtain m1block.

m1= C1⊕ IV
=1010001010111010 ⊕ 0000000000001001
=1010001010110011
For the next block of cipher previous block will serve as the input in the place of initialization vector. In this ways theblocks are chained.
C1-> C2, C2->C3...C4 to C5
Step 4: This m1 block is XORed with Armstrong number 'K' and then right shifted. Fig. 4 depicts the circular rightshift operation when r=3.
This n1 will be the input plain text first block.m1⊕ K
=1010001010110011 ⊕ 0000000010011001
=1010001000101010

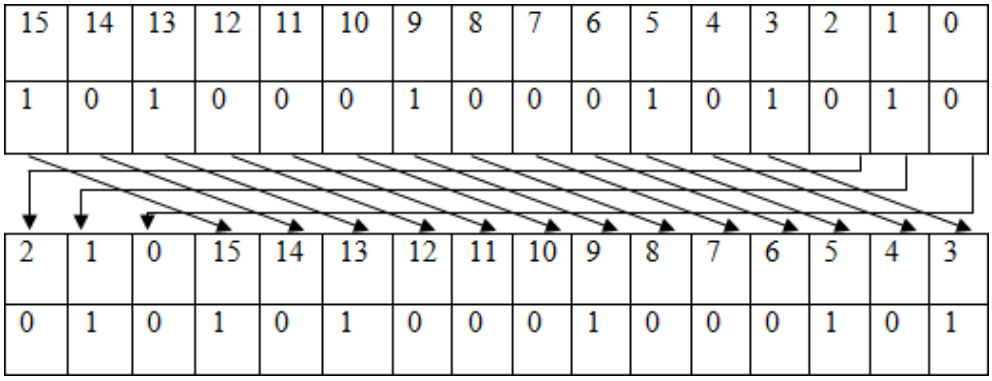

FIGURE 4. Circular right shift operations

n1=0101010001000101

For the next block of cipher, previous block will serve as the input in the place of initialization vector. In this waysthe blocks are chained.
C1-> C2, C2->C3...Cs-1 to Cs
Step 5: above steps are repeated for the remaining blocks as well, to obtain the plain text blocksn2, n3, n4 and n5.
Plain text P: 01010100010001010100110101010000001000000110010001
10110111101010000010110

CONCLUSION

This proposed technique ensures confidentiality and data integrity on binary data. For IoT things, it is very muchreliable and efficient mechanism for data communications. It very well serves the purpose of being light weight on constrained devices with little storage and power consumption. Computationally it is simple and efficient.Experimental results are evaluated on different size of plain text.

ACKNOWLEDGMENTS

I sincerely thank God for giving me the opportunity to write this paper. I wish to express my deep sense of gratitude and profound thanks to my Research Supervisor Dr. V.V. HaraGopal, Retired Professor, Department of Statistics, Osmania University, Hyderabad, Telangana, India, for his keen interest and inspiring guidance throughout the course of writing this paper. I would like to thank Loyola Academy for giving me the opportunity to present my paper at the international Multi-disciplinary conference for Emerging Technologies-2022

FUTURE SCOPE OF THE STUDY

With the increase in the size of the plain text, the binary stream size grows proportionally, so the text can be compressed using lossless technique before encryption. This further reduces the computational task and time for encryption. This algorithm can be further used to encrypt images by converting them in binary stream of data.

REFERENCES

[1] A. Eskicioglu and L. Litwin, "Cryptography," in IEEE Potentials, vol. 20, no. 1, pp. 36-38, Feb-March 2001, doi:10.1109/45.913211.

[2] A. Menezes and D. Stebila, "Challenges in Cryptography," in IEEE Security & Privacy, vol. 19, no. 2, pp. 70-73,March-April 2021, doi: 10.1109/MSEC.2021.3049730.

[3] R. Román-Castro, J. López and S. Gritzalis, "Evolution and Trends in IoT Security," in Computer, vol. 51, no. 7,pp. 16-25, July 2018, doi: 10.1109/MC.2018.3011051.

[4] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," in IEEE Access, vol. 7, pp. 82721-82743, 2019, doi: 10.1109/ACCESS.2019.2924045.

[5] Sauveron, Damien (Sauveron, Damien) *et. al.*. "Security and resilience for smart devices and applications" in Eurasip Journal on wireless communications and networking, 123, JUL 29 2014, 2014-07-29, doi 10.1186/1687- 1499-2014-123

[6] Shouran, Zaied & Ashari, Ahmad & Priyambodo, Tri. (2019). Internet of Things (IoT) of Smart Home: Privacy and Security. International Journal of Computer Applications. 182. 3-8. 10.5120/ijca2019918450.

[7] Robisson, B., Agoyan, M., Soquet, P. *et al.* Smart security management in secure devices. *J Cryptogr Eng* **7**, 47–61 (2017). https://doi.org/10.1007/s13389-016-0143-4

[8] Panahi, P., Bayılmış, C., Çavuşoğlu, U. et al. Performance Evaluation of Lightweight Encryption Algorithms for IoT-Based Applications. Arab J Sci Eng 46, 4015–4037 (2021). https://doi.org/10.1007/s13369-021-05358-4

[9] Lakshmi, M.S.; Srikanth, V.: A study on light-weight cryptography algorithms for data security in IOT. Int J EngTechnol **7**(2.7), 887–890 (2018)

[10] B. V. Varun, A. M.V., A. C. Gangadhar and P. U., "Implementation of Encryption and Decryption Algorithms for Security of Mobile Devices," 2019 IEEE 19th International Conference on Communication Technology (ICCT), 2019, pp. 1391-1395, doi: 10.1109/ICCT46805.2019.8947111.

[11] Singh, S., Sharma, P.K., Moon, S.Y. *et al.* Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *J Ambient Intell Human Comput* (2017). https://doi.org/10.1007/s12652-017-0494-4

[12] Kuzminykh, Ievgeniia & Yevdokymenko, Maryna & Sokolov, Vladimir. (2021). Encryption Algorithms in IoT:Security vs Lifetime How long the device will live?.