ISSN: 2633-4828 Vol. 5 No.4, December, 2023

International Journal of Applied Engineering & Technology

DATA ENCRYPTION STANDARD BASED AES ALGORITHM IN INTERNET OF THINGS ENVIRONMENT

P Raja Lingam[1] and Dr. Rahul Mishra[2]

[1]Research Scholar and [2]Research Guide, Department of Electronics & Communication Engineering, Dr. A. P. J. Abdul Kalam University, Indore, MP, India

[1]rajalingam.raj@gmail.com and [2]rahulmishra@aku.ac.in

ABSTRACT

Encrypting information so that it cannot be read plaintext is the focus of the field of study known as cryptography. The most popular and widely-used kind of symmetric encryption nowadays is called Advanced Encryption Standard (AES) (AES). Findings point to a speed at least six times faster than triple DES. The goal of this research is to examine how the AES algorithm stacks up against other data encryption standards used by IoT devices (IoT). We compare each technique based on how well it encrypts a sample image. Data visualisations like histograms and correlation analyses are also performed. All calculations in AES are performed using bytes rather than bits. Therefore, a 128-bit plaintext block is represented in AES as 16 bytes. These 16 bytes are organised as four columns and four rows so that they may be processed as a matrix. Therefore, the converging columns of the histogram proved that the AES approach yields higher quality data encryption. The AES algorithm also has a smaller distance to zero in its correlation coefficient, indicating a stronger link. Researchers concluded that AES is the best algorithm for protecting data on the Internet of Things (IOT).

Keywords: AES, DES, Internet of Things, encryption standard, information security

INTRODUCTION

Finding a linear connection between a subset of bits (from the message) and another subset made up of state bits that come before the replacements identified in the previous round is the main goal of linear cryptanalysis. If you want to do a linear cryptanalysis, you will need a flexible justification for the attack. It is necessary to summarise the results of the probability theory. To build and prepare these attacks in MATLAB for implementation, we need to have a solid understanding of probability theory in addition to MATLAB and C++ programming. There should be distinct chapters for linear cryptanalysis for DES implementation in C++20 and C#. MATLAB scripts may be used with examples from C++20. Check out the amazing linear assault as well as some examples of actual attacks. The most crucial methods in probability theory are the Piling-up Lemma and Linear Approximations of S-Boxes.

We'll skip over the probability tools' mathematical underpinnings for the time being due to their intricacy. Reviewing the available materials is a good idea before to implementing the MATLAB implementations. There are several similarities between differential and linear cryptanalysis. contrasting the two inputs (j) and their respective outputs One of the most important differences between differential and linear cryptography is this. The purpose of this part is not to provide you a solid foundation in mathematics. To completely understand the essential theoretical concepts, it is strongly advised that you review the listed sources. Differential cryptanalysis attacks are covered in depth in a variety of sources. Differential cryptanalysis is built on a chosen-plaintext attack. We may develop our attack plan based on the presumption that an attacker generates a large number of tuples (j, j, I I where I or value i′ is fixed.

The same secret key, K, used before is being used to encrypt the strings j and j. In 2001, the National Security Agency (NSA) and the National Institute of Standards and Technology (NIST) worked together to develop Secure Hash Algorithm 2. (SHA-2). In contrast to its successor, which offers values of 224, 256, 384, and 512 bits, the original Secure Hash Algorithm only had a set digest size of 160 bits. With far larger bit width than the previous method, it is a significant improvement. The SHA-2 algorithm seems to be collision-resistant in all of its implementations. Another bonus is that the SHA-256 hashing algorithm is used in the bitcoin sector. Secure Hash

Algorithm 3 is a new member of the SHA family of hashing algorithms developed by NIST (SHA-3). Data is fed into a sponge function in this method, which made its debut in 2015, and the outcome is then "squeezed" out. Once again, the implementation specifics of the algorithm are not the focus of this study. Keccak is another name for the SHA-3 hashing algorithm. SHA-3 has two new SHA-3 versions, SHAKE128 and SHAKE256, although SHA-2 does not. The size of the message digest is configurable and may range from 128 to 256 bits. Each SHAKE algorithm may generate a hash value that is 20, 21, or 60 bits in length. Padding in the context of cryptography refers to the practise of completing gaps in otherwise secure information. Decoding the plaintext messages in encrypted data sets is made easier by doing this. In the present period, there are several cushioning methods accessible. For instance, AES can handle a wide range of them. Here are a few of the most popular techniques that are briefly discussed. Two well-known hashing algorithms, MD5 and SHA, use a method in which they attach a hash to a block of data by appending a random number of zeros (such 1000 00).

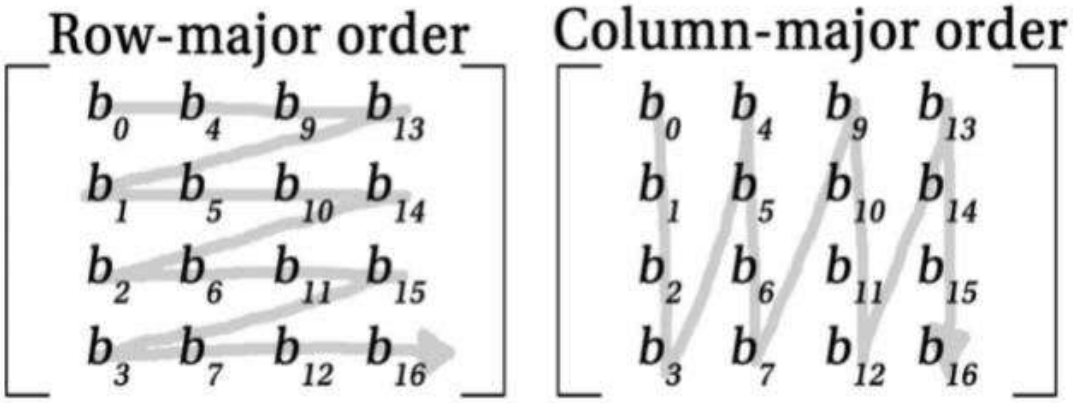

Figure 1: AES algorithm

ANSI X9.23 is a byte-sized standard that allows for the generation of random bytes to be used after each packet of data (as a reminder, one byte equals eight bits). "Zero padding" describes the efficient and aesthetically pleasing practise of adding a random number of zeros to the end of data. Padding is essential for encryption and hashing, therefore it bears repeating. The practise of salting is not limited to the food business. It's a way to make a key or password more secure by adding extra random symbols to it without the key holder knowing about it. In cryptography as compared to a single key, this provides significantly more comprehensive security. Salt strands can be as long as you like. Let's stick to hexadecimal numbers that are just 8 bits (four bytes) wide for the time being. RSA Labs, founded by Ronald Rivest, released the MD4 hashing algorithm in 1990. Windows versions 7, 8, and 10 still make use of the vulnerable MD4 hashing technique in certain aspects of user authentication. After its first introduction in 1995, MD4 was widely criticised for being too slow and too inefficient to be considered a true one-way function. In no circumstances should this hashing algorithm be used in your work. When it comes to Microsoft products, vulnerabilities using MD4 are seldom severe. Active Directory (AS) is a Windows service that handles user authentication and device authorisation. NT LAN Manager (NTLM) is a component of AS that uses MD4 hashing for authentication. In Windows, NTLM may be replaced by Kerberos, a more secure authentication system. It's only for backwards compatibility reasons that NTLM exists. On addition, in more recent Windows operating systems, Kerberos is enabled by default. The MD5 hashing method, is still frequently used today. MD5 is the fifth version of RSA Labs' hashing algorithm. When the algorithm was widely used, despite its continued popularity. This is a malicious piece of software that was initially developed for the purpose of cyber espionage. SHA-1 should be avoided because of the danger of collisions. Remember that hashing and encryption are two separate concepts. A file's hash value serves as a kind of digital fingerprint. Because they are very hard to reverse engineer, they may be used to identify datasets in a secure way. Hexadecimal numbers are used in most hash values, including the ones stated before. It is possible that no two hash values are exactly same, which is why they are so important in digital identification.

LITERATURE REVIEW

It is Y. Zhang's contention in 2019 that Data security is of paramount importance in the IoT era of widespread connectivity. However, data security in the IoT is poor. This article revises the current AES encryption standard and establishes the DESI data encryption standard to meet the needs of IoT applications (Data Encryption Standard in IoT). The results of the study indicate that DESI is the most secure method of data encryption for usage in the context of the Internet of Things.

A. S. J. Hussain Pirzada (2019): A new class of cryptographic algorithms has arisen in recent years, thanks to these advancements in encryption. In recent years, Authenticated Encryption (AE) methods have been used to offer data security services. Security for sensitive information may be provided by combining the Cipher-block Chaining Message authentication code (AES-CCM) with the Advanced Encryption Standard (AES)-CTR algorithm. However, the AES-CCM method has a limited capacity for transmitting data. As a result, this study unveils a game-changing design to boost data security throughput in communication programmes. The Cipher-based Message Authentication Code (CMAC) technique is used to send AE in the suggested algorithm. Field-programmable gate array (FPGA) implementations of both the proposed AE algorithm and AES-CCM are provided for evaluation purposes. A comparison with AES-CCM demonstrates that the suggested algorithm is more efficient in terms of processing time and throughput. It is possible to use an FPGA with a throughput of 4.30 Gbps to implement the proposed AES-CMAC AE algorithm.

T. Yue (2019): A hybrid encryption algorithm combining the high encryption efficiency of the symmetric encryption algorithm and the high encryption intensity of the asymmetric encryption algorithm has been provided after an analysis of the security weaknesses of existing WSNs. First, the MAC address and the AES key are encrypted using Elliptic Curve Cryptography (ECC), then the encryption blocks are compressed to generate ciphertext messages, and lastly, the whole ciphertext message is created by joining the ciphertext messages using data compression technology. The experimental findings of this method indicate that the stated and implemented algorithm may shorten the times required for encryption and decryption, as well as the overall running-time complexity, without sacrificing security.

RESEARCH METHODOLOGY

However, with hashing, there is a problem known as a collision, in which two values have the same hash. A number of algorithms are used to generate hash values. Ideally, a collision-free hashing method should be used. Several of them don't quite meet these standards. There are a number of algorithms that are often used for hashing applications. However, you should grasp why hashing methods are necessary for this paper aims. It's the substitution–permutation network that's at the heart of AES's robustness. For encryption, an SPN undertakes many rounds of processing. Round keys, which are generated during each round of data replacement and permutation, are also generated. To "unlock" the following round, the algorithm generates these temporary keys. Substituting one symbol for another is a simple example of substitution. The act of shuffling about the numbers in a data collection is known as "permutation." Protecting sensitive information requires more iterations of this obfuscation process.

In AES, the total number of SPN rounds is key-dependent. Using the maximum key length possible inside the AES algorithm is the safest option for encrypting sensitive data (i.e., 256 bits). A state is a 4 by 4 array where the columns are in major order; this is the data structure on which the AES works. This state's framework is organised in 16-byte cells. Storage devices like hard drives and random-access memory are linear, thus they can store two-dimensional arrays in either row-major or column-major order (RAM). Each byte of the state is then appended to a block of the round key after a call to the key schedule. The exclusive OR binary operator (XOR) is used to achieve this effect on the datasets. Assuming all other bits are 0, the result is 1 if the first or second bit is 1. In either case, the sum is 0, since 1 cannot be divided by 0. Bit values of 10011 and 11100, when XOR'd, yield 01100. Common shorthand for this service provider is the icon.

The S-box, which stands for "substitution-box," is used to swap out each of the AES state's 16 array elements with a value taken directly from a lookup table. An S-box in AES takes in one byte's worth of data (eight bits) and outputs another byte's worth of data (eight bits) using this lookup table. It is beyond the scope of this research to provide an algorithm for generating S-boxes. Assume the S-box has received the byte 0001 0010. Separate 4-bit values will be used to represent each part of this byte (i.e., nibbles). Lastly, we convert them to hexadecimal, which gives us 1 and 2. We can now use the first nibble (here, 1) to locate the appropriate record in the lookup table. The second bit (instance 2) shows AES's S-box column of interest. Our replacement number is. Sub Bytes is applied to the whole AES state during processing. The last chapter briefly explored message identification using hash values. These ideas last. We'll get into hashing after this little introduction. In 1997, the National Institute of Standards and Technology (NIST) selected the AES algorithm to replace the Data Encryption Standard (DES). This backdoor-resolving approach was supported by the cryptography community. IBM's MARS, CRYPTON, and Rijndael were among 15 algorithms submitted. AES data encryption and decryption use a single key/password, like the DES. Technology cannot crack even the weakest AES encryption until 2020. 128-bit Don't even contemplate it. The American Encryption Standard (AES) may be used commercially or non-commercially. Many popular programmes use the advanced encryption standard (AES).

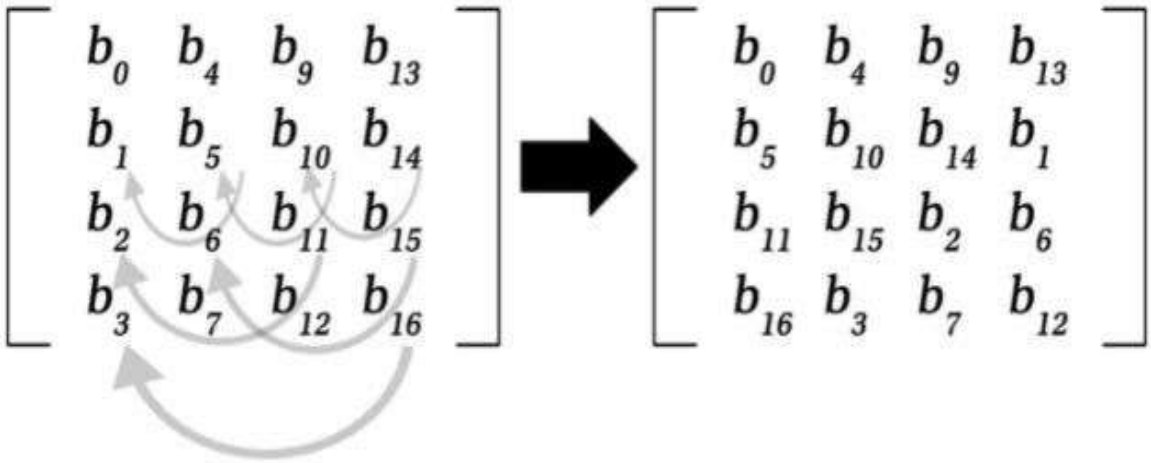

Figure 2: AES shift rows

RAR, WinZip, and 7-Zip are examples. Next chapter, we'll explore AES-based software. Modern CPUs now offer hardware-based AES acceleration from Intel and AMD, making cryptographic procedures faster than before. AES-NI is the hardware-based acceleration approach. AES-NI may accelerate cryptographic procedures tenfold under certain situations. We'll start with blocks. An algorithm works on a "block" of data in cryptography.

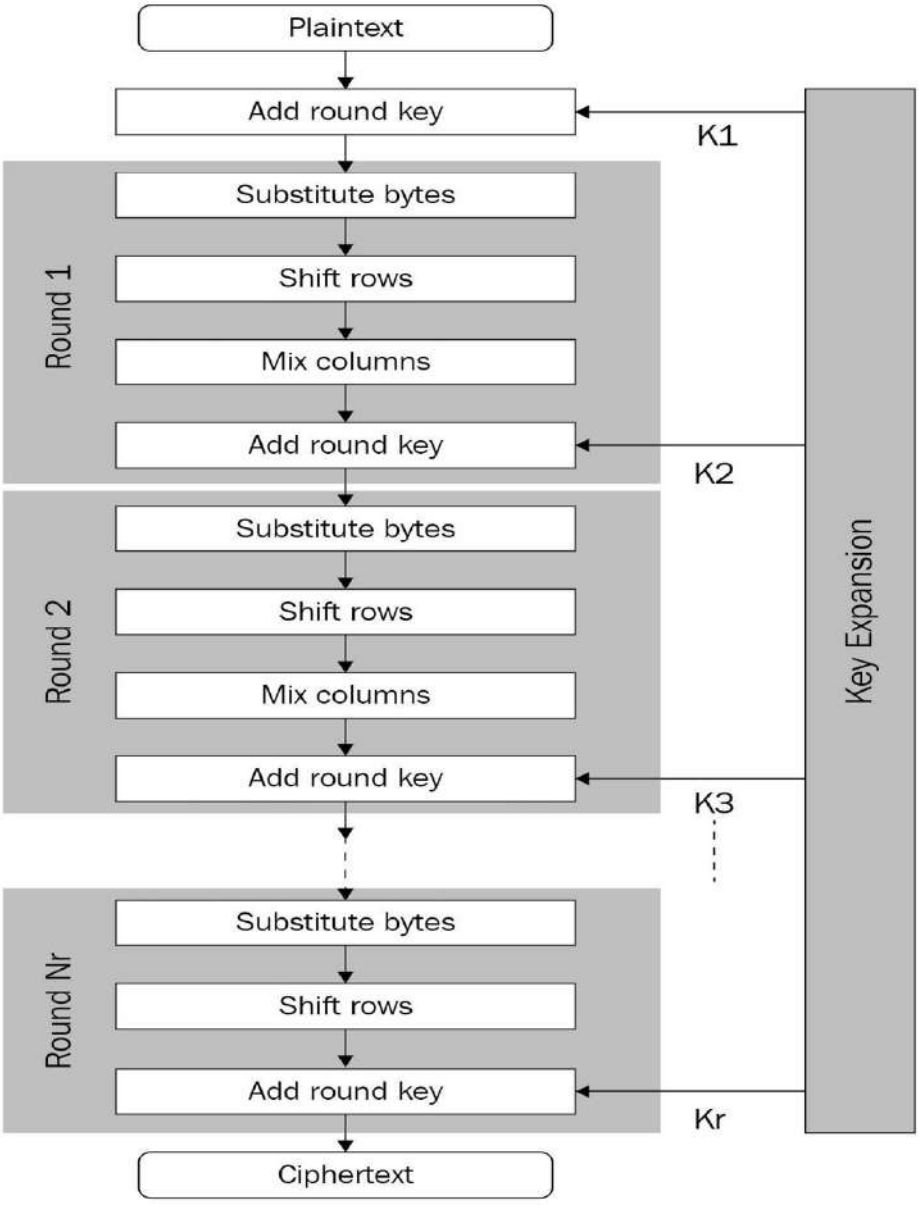

Figure 3: Block diagram of AES encryption

AES only processes 128-bit data chunks (which is the same as 16 bytes). Encrypted files may hold as many plaintext/unencrypted blocks as needed. If the plaintext fails to finish the final block, padding (usually random data) might be employed. The decimal system, sometimes known as the "base-ten" system, is universally understood. All numbers with a decimal point may be written from 0 to 9. You'll see why computers use different numbers than humans do in the next section. In the binary system, both 0 and 1 are valid symbols. The most basic operations of digital devices like computers and smartphones are conducted in a language called binary. Some binary operations often employ the values 1 and 0 for true and false, respectively. Hexadecimal, sometimes called base-16 or just hex, is the third-most-used numeric system. In addition, the letters A through F are used in hexadecimal to represent the digits 10 through 16, therefore the hexadecimal equivalent of the decimal 11 is the letter B. Hexadecimal notation, which these symbols represent, uses a total of 16 digits. The range of numbers we can call up using our two-digit Hex code is from 0 to 255.

The good news for dense information storage is that we need only employ two integers. For a binary representation, we'd need eight digits. First, let's go from decimal to binary. Let's try converting the decimal value 86 to binary and see what occurs. The concluding result is 10.10.10. Perhaps you're baffled by this development. Here's another way of looking at it: A binary number can only take on one of two values: "on" or "off." Each successive value in binary notation increases by one from the previous one. In this moment, there are two concepts that may require explaining: In a sequence of bits, the most significant bit (MSB) and least significant bit (LSB) have the highest weight. In a binary text, for instance, the first bit from the left is the most important. A binary string's last digit may represent only one (1) value. Converting from decimal to hexadecimal.

Let's turn the hexadecimal representation of the total number of crab species—67,000—into a decimal. First, we'll divide it by 16, and then by 16 again, and again, and again, until we get 0. Following each division, the remainder is written in hexadecimal notation and the digits after the decimal point are discarded. We'll need as many digits as there are stages to get there. Once the conversion is finished, the row of hexadecimal remainders is flipped. Converting from binary to hex (or vice versa) is a simple process. We will now begin the process of converting to and from hexadecimal. First, we'll follow the binary equivalents of the hexadecimal system's 16 symbols. The hexadecimal system has been replaced with a binary system using just the first four digits of a number (and the binary equivalents are always four digits). In binary form, each of the 16 hexadecimal characters. The use of DES, or Digital Encryption Standard, was rapidly becoming insecure.

Novel Algorithm:

1. The fixed key input of 128 bits is expanded into a key length depending on the size of AES: 128, 192, or 256.
2. Then, the **[K**1], **[K**2],…**[K**r] sub-keys are created to encrypt each round (generally adding **XOR** to the round).
3. AES uses a particular method called Rijndael's *key schedule* to expand a short master key to a certain number of round keys

EXPERIMENT RESULT

As computational power increased, the 56-bit keys/passwords were no longer safe enough. Although Triple DES (described in the previous chapter) is a considerably more resilient algorithm, even in hardware-based circumstances, it is still not fast enough. The National Institute of Standards and Technology (NIST) released a guideline document in 2018 stating that 3DES is being phased out. New applications will no longer be able to use the 3DES algorithm after 2023, according to these standards. On perform decryption on j and j for each tuple, the decryption algorithm is performed to the two encrypted messages. An encryption method (e.g., Diffie-DES) Hellman's may be used to decipher the plaintext that was encrypted and delivered, and the cryptanalyst can see the entire process from start to finish when they employ CPAs.

AES Encryption

Input: Novel- AES algorithm

Mode: CBC

Key Size in Bits: 128 Bits

Enter Secret Key: 1234567890123456

Output Text Format: Base64

AES Encrypted Output:

1+P6/icTFPsoiSyiD/eaIu/waIva4NAzqnRZJGUjCLd/uN8ClwmZ41m+38OmG5uyrodffCHDqPmuOh5judhLYul
UWRoEOtikOi6Z1z/eA3w=

AES Decryption

Input:

1+P6/icTFPsoiSyiD/eaIu/waIva4NAzqnRZJGUjCLd/uN8ClwmZ41m+38OmG5uyrodffCHDqPmuOh5judhLYul
UWRoEOtikOi6Z1z/eA3w=

Mode: CBC

Key Size in Bits: 128 Bits

Enter Secret Key: 1234567890123456

Output Text Format: Base64

AES Decrypted Output:

Novel-AES algorithm

DES Encryption

Input:

Novel-AES algorithm

Enter Secret Key: 1234

Output:

bwlz8uFyPSfAbLyqwuOEzFjQ3vHKXgcBX40U6ghXElW+BoEg7lSMasl4NzSWbcIqz1sY5AXSSd0i51zyuxKc
rWUTOJnwgFZGUldeXLYPpkE=

DES Decryption

Input:

bwlz8uFyPSfAbLyqwuOEzFjQ3vHKXgcBX40U6ghXElW+BoEg7lSMasl4NzSWbcIqz1sY5AXSSd0i51zyuxKc
rWUTOJnwgFZGUldeXLYPpkE=

Enter Secret Key: 1234

Output:

Novel-AES Algorithm

Cryptanalysts may pick the plaintext and encrypt active models. By selecting plaintext, the attacker or cryptanalyst may concentrate on crucial ciphertext components. Cryptanalysts must enter the "mystery box" to get the secret key. Most professional cryptanalysts and attackers have databases containing plaintexts, ciphertexts, and likely encryption keys. NIST (formerly NBS) requested cryptosystems. That day starts the interesting DES story. DES became famous worldwide after acceptance. The Lucifer algorithm inspired IBM's DES algorithm. One of the most extensive descriptions of DES is in FIPS. Feistel networks, or cyphers, underpin DES. Examine the Feistel cypher. Encryption and decryption are best done in MATLAB's ECB mode. Switching modes is quick, reliable, and effortless. Sections discuss implementation steps. These parts are simple and standardised because of the NIST standard1. The implementations were evaluated against well-known and authorised software. DES is ancient and no longer secure, and the materials supplied are enough to understand the DES algorithm's basic notions. The referenced implementations may help explain the process. Block cyphers like AES and stream cyphers like RC4 and Salsa20 are the most prevalent. Bit-by-bit processing is used. Stream cyphers blend pseudorandom data with plaintext. These cyphers enable real-time operation since they don't need all the data to decode.

The electronic code book method wastes time by constantly encrypting the same plaintext block. ECB operations need a large salting scheme. Multi-core processors may reduce system load by parallelizing data processing. ECB

use is always a bad idea. The first block uses a random seed, while the others use data from the previous block. Cypher block chaining is more susceptible to data corruption than ECB. Even a few faulty bytes in a block might compromise the data. CBC is sequential whereas ECB is parallelized. CBC will still rule in 2020 despite its drawbacks.

CONCLUSION

The AES algorithm on a 128-bit message was successfully implemented in this study. For block cyphers, we also spoke about the Diffie-Hellman and AES algorithms. This is not a comprehensive review of the mathematical foundations of block cyphers (such as DES and AES), but we have offered the most important and basic notions on which they are constructed. Discussed Feistel networks/ciphers, round, replacement keys, round, initialising vectors, math principles and terminology, etc. This chapter's most essential purpose was to provide a simple MATLAB implementation of DES and AES. Because of this, this work does not cover all elements of cryptography, both theoretical and practical. Asymmetric encryption was the subject of this work. Using a pair of keys, a public key for encryption and a private key for decryption, it's known as public key cryptography. Asymmetric cryptosystems, on the other hand, must fulfil a set of preconditions and complete a series of processes that we outlined at the beginning of the chapter. Also covered were three of the most significant classical asymmetric encryptions. In cryptography, unpredictability and primality are critical notions, hence it is necessary that the prime numbers be created using strong pseudo-random number generators and their primality be checked using robust primality tests. It is shown that the encrypted cypher text and the decoded cypher text are accurate. The suggested AES algorithm's encryption efficiency was tested, and the results were positive. There are a number of ideas that might be used in the future of this paper.

REFERENCE

1. N. Su, Y. Zhang and M. Li, "Research on Data Encryption Standard Based on AES Algorithm in Internet of Things Environment," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, pp. 2071-2075, doi: 10.1109/ITNEC.2019.8729488.
2. S. J. Hussain Pirzada, A. Murtaza, M. N. Hasan, T. Xu and L. Jianwei, "The Implementation of AES-CMAC Authenticated Encryption Algorithm on FPGA," 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET), 2019, pp. 193-197, doi: 10.1109/CCET48361.2019.8989202.
3. T. Yue, C. Wang and Z. -x. Zhu, "Hybrid Encryption Algorithm Based on Wireless Sensor Networks," 2019 IEEE International Conference on Mechatronics and Automation (ICMA), 2019, pp. 690-694, doi: 10.1109/ICMA.2019.8816451.
4. P. Mellu and S. Mali, "AES: Asymmetric key cryptographic System II", International Journal of Information Technology and Knowledge Management, vol. 4, no. 1, pp. 113-117, 2011.
5. Adi Shamir Ronald Rivest and Len Adleman, A method for obtaining digital signatures and public- key cryptosystems Communications of the ACM, vol. 21, pp. 120-126, 1978.
6. Q Zhang and Q. Ding, "Digital image encryption based on advanced encryption standard (AES)", 2015 Fifth International Conference on Instrumentation and Measurement Computer Communication and Control (IMCCC), pp. 1218-1221, 2015, September.
7. B. S Kumar, V R Raj and A Nair, "Comparative study on aes and rsa algorithm for medical images", 2017 International Conference on Communication and Signal Processing (ICCSP) (pp. 05010504), 2017, April.
8. A Al Hasib and AA MM Haque, "A comparative study of the performance and security issues of AES and RSA cryptography", 2008 Third International Conference on Convergence and Hybrid Information Technology, vol. 2, pp. 505-510, 2008, November.

9. A Chaouch, B Bouallegue and O Bouraoui, "Software application for simulation-based AES RSA and elliptic-curve algorithms", 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), pp. 77-82, 2016, March.
10. A Ray, A Potnis, P Dwivedy, S Soofi and U Bhade, "Comparative study of AES RSA genetic affine transform with XOR operation and watermarking for image encryption", 2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), pp. 274-278, 2017, October.
11. G Zhao, X Yang, B Zhou and W Wei, "RSA-based digital image encryption algorithm in wireless sensor networks", 2010 2nd International Conference on Signal Processing Systems, vol. 2, pp. V2-640, 2010, July.
12. 1.Chunling Sun, "Application of RFID Technology for Logistics on Internet of Things[J]", AASRI Procedia, no. 1, pp. 106-111, 2012.
13. Almudena D´ıaz-Zayas, Cesar A. Garc´ıa-Pe´rez and A´ lvaro M. Recio-Pe´rez, "3GPP standards to deliver LTE connectivity for IoT[C]", 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI), pp. 283-288, 2016.
14. Wang Ying, "Improvement of MixColumn() function in advanced encryption standard AES [D]", Shaanxi Normal University, 2011.
15. Christof Paar and Jan Pelzl, Exploring cryptography in depth - Principles and applications of commonly used encryption techniques [M], Beijing:Tsinghua University Press, pp. 51-111, 2012.
16. Mary James and Deepa S Kumar, "An Implementation of Modified Lightweight Advanced Encryption Standard in FPGA[J]", Procedia Technology, no. 25, pp. 582-589, 2016.