Cryptographic Algorithms for IoT Devices

Md Mokarram Alam  
Graphic Era Hill University, DehradunDehradun, India  
mdmokarram426729@gmail.com

Aditya Bhatt   
Graphic Era Hill University, Dehradun  
Dehradun, Indiaadityabhatt216@gmail.com

Abstract: The use of Internet of Things (IoT) devices has experienced a notable increase in recent years. Our project focuses on delving into this domain by investigating an IoT network that ensures the security of data through the utilization of widely recognized cryptographic algorithms, including AES, RSA, and Blowfish.

# Introduction

The Internet has consistently served as a system of interconnected networks that link computers for information sharing. Over the past two decades, a transformation has occurred in the capability to wirelessly link remote and mobile entities, including objects, utilities, or assets, to the Internet and cloud. This transformation is made possible by affordable sensors, along with rapid computing and ample storage capacities. When all these entities are interconnected, it is commonly referred to as the Internet of Things (IoT).

The International Energy Research Centre (IERC) defines IoT as follows :

" A dynamic global network infrastructure, capable of adapting itself, employs standardized and compatible communication protocols. Physical and virtual entities within this system have unique identities, physical traits, and virtual characteristics. They communicate through intelligent interfaces and seamlessly become part of the information network." (1)

NEED FOR CRYPTOGRAPHIC ALGORITHMS

The primary issue associated with a vast network comprising various device types is security. More specifically, there is a concern for personal privacy, as these devices have the potential to expose sensitive information and create security vulnerabilities. To address this, cryptographic solutions, particularly ciphers, are employed to ensure data integrity, user authentication, and confidentiality. While numerous cryptographic methods exist for ensuring IoT security, our research focuses on ciphers, which we also term cryptographic algorithms.

A cryptographic algorithm functions as a tool for encoding information. It takes inputs such as plaintext (data that is both easily understood and readable, whether it is in a stationary or transitional state.) and a key (such as a password), and generates ciphertext (scrambled data in stationary or transitional state)

as the output, particularly in IoT environments. The primary function of these algorithms is to secure and unsecure information.

* 1. Cryptographic Algorithms

Cryptography is the method of securing information from unauthorized access, exploitation, disclosure, interception, alteration, or destruction. Its purpose is to protect data both during transmission and while at rest.

There are three main types of cryptography algorithms: symmetric, asymmetric, and cryptographic protocols.

• Symmetric algorithms

These algorithms use a single key for encrypting and decrypting data. The key must be known by both the sender and the recipient for secure communication. Some examples of symmetric algorithms include AES, DES, RC4, Blowfish and 3DES.

• Asymmetric algorithms

These algorithms operate with two keys: a public key and a private key. While the public key is employed for encrypting data, only the private key possesses the capability to decrypt it. This allows anyone to encrypt data for a specific recipient, but only the recipient can decrypt it. Some examples of asymmetric algorithms include RSA, DSA, and ECC.

• Cryptographic protocols

These are the union of symmetric and asymmetric algorithms to achieve specific security goals, such as secure authentication and key exchange. Some examples of cryptographic protocols include SSL/TLS, HTTPS, and SSH.

Cryptographic algorithms can be implemented in hardware or software. Hardware implementations are typically faster and more secure, but they can be more expensive. Software implementations are more flexible and can be deployed on a wider range of devices, but they may be slower and less secure.

* 1. IoT Architecture

IoT devices have unique identifiers (IP addresses or URIs) and can be remotely observed, controlled, managed, and configured. They are linked to an information network to transmit data with other systems and devices. This enables them to recognize and be recognized by other devices in the network, share information, and collaborate more efficiently.

Kevin Ashton from MIT proposed the concept of IoT in 1999, based on RFID technology. Presently, a variety of sensing devices, like Radio-Frequency Identification, ZigBee, Infrared, GPS, Wi-Fi, are connected to the internet. These devices employ various protocols for communication and information exchange.

IoT systems consist of hierarchical layers, including the application, network, and the perception layer. Every layer employs its own mechanism, and devices within the given layer that may utilize different technologies. While this setup offers a broad range of services in the network, it also poses challenges in managing diverse devices and technologies. The IoT architecture comprises several layers, with the application layer serving as the outermost and most visible layer. Its primary purpose is to meet industrial demands and fulfill intellectual needs by performing various functions. It takes in the information from the middle layer and processes it to provide high-standard favor to the users.

The middle layer is liable for taking the data from the network layer, processing and storing device information in the cloud and databases, and supplying the necessary APIs to the primary layer(AL).

The network layer is essential for establishing the infrastructure needed for IoT. It collects data from the perception layer and passes it via the wired medium or wireless medium to the middleware layer. Technologies like

WiFi and Bluetooth are utilized for transmitting data. Furthermore, this layer plays a crucial role in protecting against widespread attacks that can disrupt device coordination and information sharing.

Within the perception layer, objects are recognized, and their data is gathered before undergoing conversion into digital signals.

**Different Layers of IoT Architecture**

2. CLASSIFICATION OF CRYPTOGRAPHIC ALGORITHMS

*Symmetric Algorithm*

1. DES

DES imply Data Encryption Standard. The DES algorithm uses a 56-bit key. With this key, DES enters 64-bit plain text blocks and generates 64-bit encryption text blocks. The DES process has several ways to participate in it, and each step is called a round. The number of rounds varies according to the size of the key utilized. For example, 128 bit crucial requires 10 rounds,192 bit crucial requires 12 rounds, etc.. The original modification or IP processes only occurred prior to the 1st round. This is the process of adapting. It is deceives the bit location of the main text block sufficiently. For instance, the IP procedure substitutes the 58th bit of the primary text block for the 1sr bit, the 50th bit for the alternative bit, the 42nd bit for the 3rd bit, etc. When the IP technique has been completed, the64-bit text blocks that are changed are split into 2 semi-blocks, containing 32-bits each. The 2 semi-blocks are LPT and RPT (left and right plain text respectively).

Before the procedure starts, each 8th bit of the 64-bit DES element is excluded. Therefore, the excluded bit positions are 8, 16, 24, 32, 40, 48, 56, and 64,making them crucial for 56 bits. Now, the key transformation involves the generation of a subkey with 48-bit spot of 56-bit size of key. There are two halves of the critical 56-bits, and each half has a 28-bit entity. Both of these parts rotate by one or two positions, depending on the count of rounds. For instance, for the rounds first, second, ninth and sixteenth, each of the 2 given halves move to a single position. The leftover bones are moved by two. This procedure generates a 48 crucial bits. During the relocation process, some of the 56 bit components are misplaced or lost. Consequently, it forms the key of 48 bit size. This procedure is known as contraction arrangement. Following this, we had two sections – Left Plain and Right Plain Text. The RPT is increased from 32-bit to 48-bit throughout this operation. Initially, the 32-bit RPT is split up into eight blocks. Each and every block contains 4 bits. In addition, there are two redundant bits added to each four-bit block that operates in six-bit blocks. Then, the bits are processed in a process of permutation to create 48-bit data. Furthermore, 48-bit RPT and 48-bit critical data gathered during the compression and arrangement process using XOR properties or function.

1. AES

AES stands for Advanced Encryption Standard. It is interesting to note that the AES execute every calculations on bytes instead of bits. Thus, AES handle plain text of 128 bits as 16octets. In order to process these 16 bytes as a matrix, they are arranged in 4 rows and 4 columns.

The number of rounds in AES varies and is dependent on the length of the key, unlike DES.

In the AES encryption method, we use different size of key:128-bit, 192-bit, and 256-bit, and set the number of rounds in the process. There are ten rounds in a 128-bit key, twelve in a 192-bit key, and fourteen in a 256-bit key. The original AES key is used to calculate a distinct 128-bit round key for each round.

The AES architecture is as follows:

As described in the design, the first 16 input bytes are converted using a fixed table called the S-Box. The result forms a 4x4 matrix.

Each row of the matrix is converted to the left, and any element outside the boundary is re-introduced to the right side of the row. Changes occur as follows:

* The initial row does not change.
* A space is added to the left by the second row.
* The third row shifts two places to the left.
* Three places to the left are made up by the fourth row.

This results in a new matrix where the 16 bytes are reorganized.

Next, each four-byte column is subject to certain mathematical transformations. This transformation uses the 4 bytes of the column as input and creates a entirely new data bytes to replace the inaugural column. This process generates auxiliary matrix containing 16 bytes of new data. It is important to know that the final encryption round will not include this step.

The matrix’s 16 bytes are treated as 128 bits and combined with 128 bits of round keys by XOR. If it is the final round, a cipher text is produced as output. Otherwise,128 bits are translated into 16 bytes and we start over with a similar rounds.

In today's cryptography landscape, AES enjoys widespread adoption and is supported across various hardware and software platforms. As of now, there have been no empirical cryptographer strike discovered in adverse to AES. Furthermore, AES offers inherent flexibility in key length, offering a form of future-proofing against advancements in exhaustive key search capabilities.

However, like DES, the security of AES is only guaranteed if it is executed properly and its key management is used efficiently.

1. Blowfish

Blowfish, designed by Bruce Schneier in 1993, is a versatile 64-bit symmetric key block algorithm with a variable-length key. Its creation aimed to provide a swift and cost-free substitute for the outdated DES and IDEA encryption techniques.

Blowfish surpasses DES and IDEA in terms of speed and is freely available for various applications, as it is not subject to patents. However, it couldn't fully replace DES because of its small and considered insecure 64-bit block size. Its successor, Twofish ,addressed this security concern by using a larger 128-bit block size. Nonetheless, Blowfish remains unbroken in terms of full encryption and is integrated into numerous encryption products and cipher suites used today.

The block size used by Blowfish is 64-bit and, varying length keys with a length of 32 to 448 bits are supported. The algorithm involves a sixteen rounds Feistel network, with each round operating on 64-bit block distributed into two words of 32 bit. It encrypts and decrypts data using the same encryption key.

Data encryption in Blowfish involves a series of 16 Feistel-like iterations, where each round includes key dependent arrangements as well as key reliant substitutions. Data encryption process in Blowfish relies on significant key-dependent S-boxes, which are critical for the substitution process. All encryption operations are based on XOR and addition operations on 32-bit words.

In the key expansion, we convert the maximum of 448-bit keys into several subkey arrays of 4,168 bytes each. These subkeys play a vital role in the Blowfish algorithm. The subkeys are precomputed prior to the encrypting or decrypting process. In Blowfish, the B-array has eighteen 32-bit subkeys and 4 S-boxes of 256 entries.

1. RC4

The RC4 encryption algorithm, also known as Rivest Cipher 4, is a symmetric key encryption method used to encrypt small portions of plaintext data, creating a stream cipher. This encryption technique establish the protected transmission of private information on websites and is widely used in various platforms, including TLS. RC4 algorithm consists of 2 main components:

KSA (Key Scheduling Algorithm) and the PRGA (Pseudo-Random Generation Algorithm). Together these two methods produce the stream cipher. RC4 produces a pseudo-random bit stream, which is called a keystream. Similar to other stream ciphers, this keystream can be utilized for data encryption by applying bitwise exclusive OR (XOR) with the plaintext, and decryption of data is processed in the same way, making use of XOR. This process is similar to the concept of a 1-time pad, besides that RC4 generates pseudorandom bits stream instead of predefined bit stream.

In order to form stream of bytes, the RC4 algorithm relies on a concealed cognitive state comprised of 2 elements: an arrangement denoted as P, encompassing all 256 bytes, and 2 index pointers of size 8-bit designated as x and y. The arrangements undergoes initialization through the KSA algorithm, using a varying length of key ranging from 40 to 2048 bits. Following this setup, the bitstream is generated using PRGA.

A lot of stream encodings rely on LFSRs, which are fast in hardware but slow in software implementations. RC4’s design does not rely on LFSR and is well suited for software implementation because it mainly involves manipulating bytes.

To reduce a value by a factor of 256, perform bitwise AND with 255, which is the same thing as getting the bytes of the lower order of the given data.

*Asymmetric Algorithm*

1. ECC

ECC stands for Elliptical Curve Cryptography. ECC, a substitute of RSA, is a robust cryptographic method that establishes secure pairs of keys used in public key encryption by leveraging the principles of elliptic curves. While RSA achieves a similar outcome using prime numbers instead of elliptic curves, ECC has attained popularity due to its smaller key sizes and strong security, and this pattern is anticipated to persist as the need for secure devices continues to increase., especially in the context of limited mobile resources. Understanding elliptic curve cryptography is crucial in this context.

In divergence to RSA, ECC builds its public key cryptographic investigation into the algebraic characteristics of elliptic curves within finite fields. This results in ECC keys that are more mathematically challenging to break, making ECC the advance solution for public key cryptographic algorithm and a more secure alternative.

The adoption of ECC also makes sense for achieving both high performance and security. As more websites prioritize online security for customer data and mobile optimization, ECC is increasingly being used. This growing usage of ECC for data security underscores the need for accessible guides on elliptic curve cryptography. The goal is to tackle issues related to access control, collective authentication, and privacy by hiding confidential data and user identity details.

In the context of current Elliptic Curve Cryptography (ECC) applications, an elliptic curve refers to a two-dimensional curve defined over a finite field. This curve is composed of points that adhere to a particular equation.

*a²=b³ + cx + d.* (Eq.1)

In ECC-based cryptography, a specific elliptic curve and two parameters, a and b are chosen within a finite field 'Fp' such that the equation 4a3 + 27b2 ≠ 0 (mod p) holds. This ensures that the elliptic curve is valid. Points (x, y) that satisfy this equation belong to the elliptic curve, denoted as E in the field Fp.

For encryption and decryption, a private key x (1, l-1)is randomly selected from the field Fp. The public key Pk is determined as Pk = PnA, where A is any point present on the elliptical curve, and Pn is the key which is private. Every entity is then transferred into bits that define (a, b) points lies on the elliptical curve. These points are enciphered and further transformed into bits.

The ECC encryption process involves the following steps

* Initialization: Both sides in the communication choose the elliptic curve E and a point A with a certain order p.
* Generation of public key: The public key Pk is determined as Pk = PnA. Pu is accessible to both the sender and the recipient and the key Pn private key needed for decryption.
* Encrypting: An arbitrary number n is chosen, and the data is encrypted with the help of a specific equation (Eq. 2). The sender utilizes the result R to send the message d to the recipient.

*R = Enc(d) 🡪 (a1,a2)* (Eq.2)

*Where R is cyphertext, d is message or data c1 = nG, and c2 = d + nPk.*

* Decrypting: Upon receiving R, the recipient determines the original data using the equation (3).

*Dec(R) =a 2 - Pk..a1 = d +nPkA = d* (Eq.3)

Overall, ECC-based encryption involves selecting appropriate curve parameters, generating public and private keys, and encrypting messages using the chosen elliptic curve.

1. RSA

RSA, a cryptographic system developed by Rivest, Shamir, and Adleman in 1977. This is a form of public key encryption and digital signature technology. It operates using two distinct keys: a Public Key, shared publicly or openly, and a secret Private Key.

An example of how asymmetric cryptography, such as RSA, works is as follows:

* A client (such as a web browser) sends data to a server using its public key.
* After using the client’s public key to encrypt the data, the server sends it to the client.
* The client then decrypts and receives the encrypted data using its own private key.

The foundation of RSA is the difficulty of factoring big numbers. The public key is made up of 2 numbers, among them one is the result of the multiplication of two big prime numbers, and the private key is also made up of these prime numbers. If that big number is able to factorized, the private key can be threatened. The potential of RSA encryption is determined by the length of the key. A larger key (maximizing the size) increases the encryption potential exponentially. The key length of RSA are either the size of 1024 or 2048 bits. Specialists warn that a 1024-bit key may soon become unbreakable. However, as of now, this remains a highly challenging task. The RSA algorithm consists of four main phases:

* Key Generation (First Stage): During this phase, a private key which is kept secret and a public key (shared with others) are created.
* Key Distribution (Second Stage): Here, the public key is widely dispersed across the network.
* Encryption (Third Stage): The sender enciphers the information using the public key of recipient.
* Decryption (Fourth Stage): The receiver deciphers the information using its private key to retrieve the original content

In the procedure of encrypting and decrypting the bits of information, a public key (a, k) and a private key (b, k) are used as a positive integers. The process involves the following steps:

*Enc(D) =Da mod k = C, where D = Message or Data, C=Cyphertext*

*Dec(C) = Cb mod k*

*k = multiplication of two prime numbers l and m (k=l\*m)*

*a = huge random contingent prime to l*

*b = multiplicative inverse of modulo(l-1)\*(m=1)*

Only the browser itself cab decrypt the data in an asymmetric system like RSA, even if a third party possesses the browser’s public key.

1. ABE

ABE, which stands for attribute-based encryption, is a type of public key encryption that lets users encrypt and decrypt messages according to user entities. In a standard ABE system, the length of the encrypted message grows with the addition of attributes, and the decryption time increases as more attributes are involved in the decryption process.

KP-APE and CP-APE are the two primary categories of encryption schemes based on the attribute

In KP-APE, secret keys of user are created which is established on an access tree, which explains the user’s entitlement and extent of access. The message is enciphered using a collection of attributes. In CP-APE, data is encrypted using access trees, and user secret keys are generated on the basis of a collection of attributes.

CP-ABE (2) encryption is a complex process that is not accomplished by data resources. Instead, it is performed by the ABE proxy. Furthermore, additional factors like the services provided by IoT cloud platforms are considered for improving, storing and reclamation of data, along with the management of AES enciphered symmetric keys. For example, the KGS uses the consumer’s data in the form of attributes from the profile to give the ABE server proxy the message it needs to encrypt the data.

A similar approach was taken with the introduction of the secure smart health system (SSH), which is based on ABE (3). The SSH system includes reduced aggregate signatures and anonymous certificates, as well as an anonymous CP (Access Control, Aggregated Authentication, and Privacy) scheme made up of 4 algorithms.

**CP-ABE architecture for IoT platform**

D. ABS

A new EPASS (signature scheme) has been proposed for an Advanced ABS system that utilizes an attribute tree and employs strategies involving AND, OR gates based on the calculative Diffie-Hellman problem (4). Regarding the Internet of Things (IoT) context, where connections are ubiquitous and user privacy is vulnerable, it is essential to have authentication methods that provide control over user attributes. As a result, there’s a need for an evolving signature methods based on attribute that takes user private information and attribute policies into account. Rising signature methods which is based on attributes empower resource requesters to generate signatures that satisfy specific attribute policies without revealing excessive details. However, only a few current methods simultaneously achieve a well-defined strategy and security under the traditional Diffie-Hellman principle. These signature schemes prevent users from counterfeiting signatures using attributes one does not possess, and they assure that a document can only be endorsed by someone possessing the necessary attributes to fulfill the protocol, ensuring non-repudiable capabilities. Authoritative signers remain beloved and distinct from all users whose attributes comply with the supervision, safeguarding the solitude of attribute-holding signers. Their solution has shown improved efficiency compared to previous systems, granting at the cost of increased computation and larger signature sizes.

**Performance Analysis of Different Algorithms**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Algorithm** | **Key Size (bits)** | **Average Encryption time (ms)** | **Average Decryption time (ms)** |
| 1 | AES | 256 | 300 | 293 |
| 2 | Blowfish | 128 | 287 | 278 |
| 3 | DES | 56 | 292 | 282 |
| 4 | RC4 | 64 | 283 | 280 |
| 5 | RSA | 1024 | 541 | 450 |

*References*

1. European Research Cluster on the Internet of Things, “ Internet of Things,” European Research Cluster on the Internet of Things, [Online]. Available: http://www.internet-of-thingsresearch.eu/about\_iot.htm. [Accessed May 3, 2018].
2. 69. Pe´rez, S., Rotondi, D., Pedone, D., Straniero, L., Nu´n˜ez, M. J., & Gigante, F. (2018). Towards the CP-ABE application for privacy-preserving secure data sharing in IoT contexts. In Innovative mobile and internet services in ubiquitous computing. Cham: Springer International Publishing.
3. Zhang, Y., Deng, R. H., Han, G., & Zheng, D. (2018). Secure smart health with privacy-aware aggregate authentication and access control in the Internet of Things. Journal of Network and Computer Applications, 123,89–100. https://doi.org/10.1016/j. jnca.2018.09.005.
4. Su, J., Cao, D., Zhao, B., Wang, X., & You, I. (2014). ePASS: An expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the Internet of Things. Future Generation Computer Systems, 33, 11–18. https://doi.org/10. 1016/j.future.2013.10.016.
5. 3. Mousavi, S. K., Ghaffari, A., Besharat, S., & Afshari, H. (2020). Improving the security of Internet of things using cryptographic algorithms: A case of smart irrigation systems. Journal of Ambient Intelligence and Humanized Computing. https://doi. org/10.1007/s12652-020-02303-5.
6. [*https://www.researchgate.net/publication/317258631\_Performance\_Analysis\_of\_Different\_Cryptography\_Algorithms*](https://www.researchgate.net/publication/317258631_Performance_Analysis_of_Different_Cryptography_Algorithms)*.*
7. [*https://link.springer.com/chapter/10.1007/978-3-642-36362-7\_11*](https://link.springer.com/chapter/10.1007/978-3-642-36362-7_11)
8. An analysis and a comparative study of cryptographic

algorithms used on the Internet of Things (IoT) based on

avalanche effect by Khumbelo Difference Muthavhine.