IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Comparative analysis of Security Algorithms for Internet of things

[1]Ansari Maqsood Ahmed, [2]Dr. Prasadu Peddi, [3]Dr. Pathan Mohd. Shafi

[1]Research Scholar, Shri JJT University, Jhunjhunu, Rajasthan,
[2]Assistant Professor, Dept. Of Computer Engineering, Shri JJT University, Jhunjhunu, Rajasthan,
[3]Professor, MITSCO, MITADT University

Abstract : The Internet of Things (IoT) is most important area that will connects huge number of various devices with different operating systems and different capabilities. The huge nature of the data in this world gives rise to the security threats to the entire system. Security of the Internet of Things provides an area that focuses on academia and industry as well. In this area, authentication and authorization methods play a key role a. In this paper, we discuss various authentication and authorization approaches for IoT devices. These encryption algorithms that are mathematically bit expensive due to the limited potential of IoT devices. We provide a detailed analysis through that we conclude the optimum security algorithm best suited for IoT devices. In this paper, we have gone through different encryption mechanism and provide a comparative analysis. This analysis lay the foundation for the future studies and research**.**

Index Terms **- Big data security; public cloud; security threats; security vulnerabilities, Security in IOT devices; Lightweight algorithm, Security algorithm for IOT devices, authentication and authorization approaches.**

I. INTRODUCTION

The Internet of Things (IoT) is advancing at a lightning speed. By 2022, it is expected that the number of IoT physical devices will reach 42.62 billion, contributing to economic growth worldwide. Every facet of our life is being impacted by the Internet of Things, including smart health, smart living, smart supply chains, smart manufacturing, and smart agriculture. In short, the existing benefits of the Internet of Things are huge, and they are certain to expand in the near future as more creative technologies emerge. IoT, on the other hand, will be the primary internet consumer, with a compound annual growth rate (CAGR) of 145 % projected through 2025. It has become a particularly attractive target for fraudulent users, and it is open to a variety of attacks, thanks to its rapid expansion [15].

These devices can connect with one other as well as with the local network or the Internet. The data transmission procedure must be safeguarded since they frequently acquire sensitive and, in some circumstances, private information. IoT devices have limited computing power, physical memory, and power supplies. The purpose of this study is to adopt a structured framework to measuring operational delay by running commonly used cryptographic algorithms on current data. We also explore the suitability of algorithms when system resources are constrained, as well as concerns for optimal performance, energy efficiency, and application needs [1].

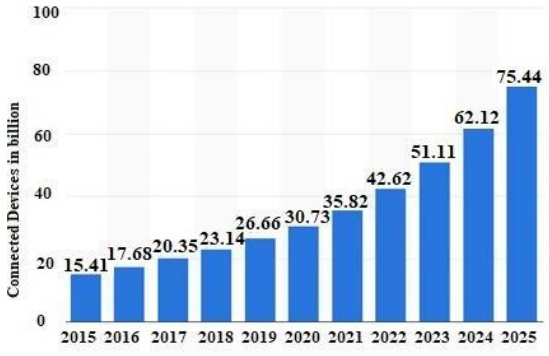

Fig. 1.1: Internet of Things (IoT) connected devices from 2015 to 2025 (in billions)

For providing security, authentication, or encrypting payload data in a transmission, several cryptographic methods have been developed. However, not all of the methods can be used in IoT devices with limited resources. It is critical to understand what a hardware platform offers and how well a given algorithm operates on it when using cryptographic methods. We examined variety of lightweight cryptographic methods to see how they performed and what advantages and disadvantages they had[3].

II. CONTRIBUTION OF THE PAPER

This paper aims to contribute a better understanding of security algorithm performance by examining the performance of various algorithms that can be implemented on IoT devices. It does so by comparing and analysing the performance of a variety of security algorithms, such as authenticated encryption schemes, AES, block cyphers, message authentication codes, hash functions, and elliptic curves, on an IoT platform.

III. OVERVIEW OF VARIOUS CRYPTOGRAPHIC ALGORITHMS

1) Lightweight Cryptography
2) Diffie-Hellman Key Exchange Algorithm
3) RSA Algorithm
4) Advanced Encryption Standard Algorithm (AES)
5) Elliptic Curve Cryptography (ECC)

1. LIGHTWEIGHT CRYPTOGRAPHY

Lightweight cryptography is designed for a wide range of devices and may be implemented on a number of hardware and software platforms. Servers and desktop computers are on the upper end of the device range, followed by tablets and smartphones. Because conventional cryptography algorithms may work effectively in these devices, lightweight methods may not be required. Devices like embedded systems, RFID devices, and sensor networks are at the lower end of the spectrum. The focus of lightweight cryptography is on the extremely limited devices found at the lower end of this range.

There is a trade-off between performance and resources required for a particular security level in cryptographic algorithm design. Power and energy usage, latency, and throughput are all ways of measuring performance. Gate area, gate equivalents, and slices are commonly used to describe the resources required for a hardware implementation. This is mirrored in the use of registers, RAM, and ROM in software. Because adding additional gates or memory tends to raise a device's production cost, resource needs are frequently referred to as costs[7].

Because of the nature of many limited devices, power and energy consumption are important parameters. In gadgets that gather power from their atmosphere, power may be very important. An RFID chip that uses the electromagnetic field transmitted by a reader to power its internal circuit is a good example. In battery-operated systems with a set quantity of stored energy, energy consumption (i.e., power consumption over time) is very significant. Some gadgets' batteries may be difficult or impossible to recharge or replace once they've been deployed. It's also worth noting that power consumption is influenced by a variety of parameters, including the threshold voltage, the clock frequency and the technology used for implementation.

Latency is extremely important in some real-time applications, such as automobile applications where components like steering, airbags, and brakes must respond quickly. It may be defined as the time it takes for an operation to complete from the moment it is requested to the time it is completed. The delay between the first request for encryption of a plaintext and the reply that provides the appropriate ciphertext is an example of encryption latency.

The pace at which fresh outputs (such as authentication tags or ciphertext) are generated is known as throughput. High throughput may not be a design aim in lightweight architectures, unlike traditional primitives. However, average throughput is still required in most applications[9].

2. DIFFIE-HELLMAN KEY EXCHANGE ALGORITHM

DH is a mathematical process that allows two devices to generate an identical shared secret on both systems, even if they have never communicated with one another. The shared secret may then be used to exchange a cryptographic encryption key in a secure manner. The traffic between the two systems is then encrypted using this key.

Diffe-Hellman is not an encryption system as we usually think of it, because we don't use it to encrypt data. Rather, it's a method for exchanging keys that encrypt data in a safe manner. This secure exchange is carried out via DH by establishing a "shared secret" (also known as a "Key Encryption Key" or KEK) between two devices. The shared secret then encrypts the symmetric key for secure transmittal. The symmetric key is some of the time called a "Traffic Encryption Key" (TEK) or "Data Encryption Key" (DEK)[19].

When each of the two users A and B creates a private key, the operation begins. After that, each user generates a public key that is a derivation of the private key. The public keys of the two users are then exchanged. Both users A and B now have their own private key and the public key of the other user. The operation continues once the key exchange is completed. The Diffie-Hellman protocol generates a "shared secret," a cryptographic key that is same for both users.

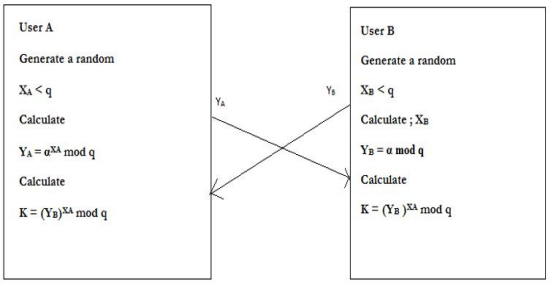

Fig 1.2 : Diffie-Hellman Key Exchange

You generate a value by performing the mathematical operation on your own private key and the public key of the other side. When the remote end performs the identical operation on your public key and its own private key, it likewise generates a value. The important thing to remember is that the two characteristics created are identical. They are the "shared secret" that allows systems to encrypt data.

3. RSA ALGORITHM

The RSA algorithm is a public key encryption technology that is often regarded as the most secure. Rivest, Shamir, and Adleman devised the RSA algorithm in 1978, thus the name. This is an asymmetric cryptography algorithm, meaning it employs both a public and a private key (i.e. two different, mathematically linked keys). A public key is shared openly, but a private key is kept private and must not be shared with anybody.

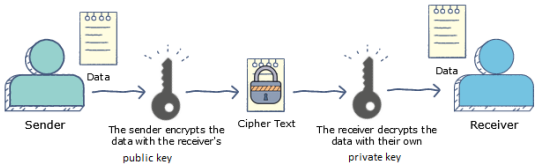

Fig. 1.3 : RSA Algorithm Working

The RSA concept is predicated on the fact that factoring a big number is difficult. The public key is made up of two integers, one of which is the result of multiplying two huge prime numbers. The same two prime numbers are also used to create the private key. As a result, if the huge number can be factored, the private key is compromised. As a result, encryption strength is entirely dependent on key size, and as key size is doubled or tripled, encryption strength grows exponentially. RSA keys are normally 1024 or 2048 bits long, however experts fear that 1024 bit keys may soon be broken. However, it appears to be an impossible feat at this time.

RSA algorithm is stronger than any other symmetric key algorithm, and the benefits of the RSA algorithm in cryptography are authenticity and privacy[4].

4. ADVANCED ENCRYPTION STANDARD ALGORITHM (AES)

The Advanced Encryption Standard (AES) is the most popular and commonly used symmetric encryption algorithm available today (AES). It is at least six times quicker than triple DES in terms of discovery.

Rather than being a Feistel cypher, AES is an iterative cypher. It is built on the basis of substitution permutation network. It consists of a sequence of connected processes, some of which require substituting specified outputs for inputs (substitutions) and others involving shuffling bits about (permutations).

Surprisingly, AES uses bytes rather than bits for all of its calculations. As a result, AES considers a plaintext block's 128 bits as 16 bytes. For matrix processing, these 16 bytes are organised into four columns and four rows.

In contrast to DES, the number of rounds in AES is configurable and dependent on the key length. For 128-bit keys, AES employs 10 rounds, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Each of these rounds use a unique 128-bit round key derived from the original AES key.

A typical round of AES encryption is seen below. There are four sub-processes in each round. The following diagram depicts the first round of the process:

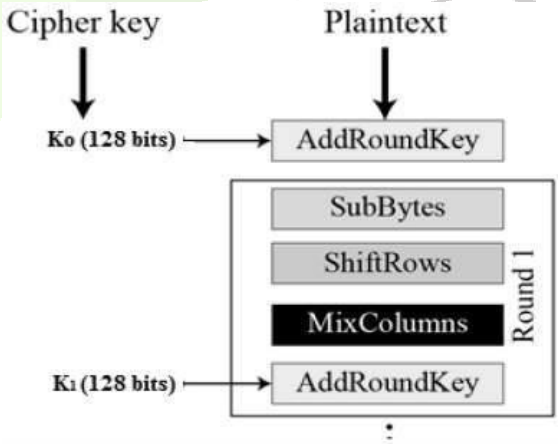

Fig. 1.4 : Encryption Process

By searching up a fixed table (S-box) provided in design, the 16 input bytes are replaced. The end result is a four-row, four-column matrix. Each of the matrix's four rows is moved to the left. Any 'falling off' entries are re-inserted on the right side of the row.

A particular mathematical function is now used to alter each column of four bytes. This method takes four bytes from one column as input and returns four entirely new bytes that replace the original column. As a consequence, a new matrix with 16 additional bytes is created. It's worth noting that this stage is skipped in the final round.

The matrix's 16 bytes are now treated as 128 bits, and they are XORed with the round key's 128 bits. The ciphertext is the output if this is the final round. Otherwise, the 128 bits are interpreted as 16 bytes, and the process repeats again.

The decryption of an AES ciphertext is identical to the encryption process in reverse order, with the exception that for a Feistel Cipher, the encryption and decryption algorithms must be implemented independently, despite their close relationship.[17]

5. ELLIPTIC CURVE CRYPTOGRAPHY

ECC is a public-key cryptography technique based on the algebraic structure of elliptic curves over finite fields. In comparison to non-EC cryptography (based on ordinary Galois fields), ECC allows for fewer keys to give equal security. Key agreement, digital signatures, pseudo-random generators, and other jobs can all benefit from elliptic curves. By combining the key agreement with a symmetric encryption algorithm, they may be utilised for encryption indirectly. Elliptic curves are also utilised in numerous elliptic-curve-based integer factorization techniques with cryptographic applications, such as Lenstra elliptic-curve factorization.

An elliptic curve, for contemporary cryptographic applications, is a plane curve over a finite field (rather than the real numbers) that consists of points satisfying the equation:

$$y^2 = x^3 + ax + b$$

Along with a distinct point at infinity, marked. The curve equation will be slightly more difficult if the coordinates are taken from a fixed finite field with a characteristic not equal to 2 or 3. This set is an abelian group, with the point at infinity as an identity member, when combined with the group operation of elliptic curves.

Several discrete logarithm-based methods have been extended to elliptic curves, as follows:

- The Diffie–Hellman key agreement system is based on the Elliptic Curve Diffie–Hellman (ECDH) key agreement mechanism.
- The Elliptic Curve Integrated Encryption Scheme (ECIES), also known as the Elliptic Curve Augmented Encryption Scheme or simply the Elliptic Curve Encryption Scheme, is a cryptographic scheme that uses elliptic curves to encrypt data.
- The Digital Signature Algorithm (DSA) provides the foundation for the Elliptic Curve Digital Signature Algorithm (ECDSA).
- Using Harrison's p-adic Manhattan metric as a deformation technique,
- The Edwards-curve Digital Signature Algorithm (EdDSA) employs twisted Edwards curves and is based on the Schnorr signature.
- The ECMQV key agreement scheme is based on the MQV key agreement scheme.
- ECQV implicit certificate scheme is based on the MQV implicit certificate scheme.
- Elliptic curves can be used for encryption, digital signatures, pseudo-random generators, and a variety of other applications. They're also employed in a number of integer factorization methods with cryptographic applications, such as the Lenstra elliptic-curve factorization[5].

IV. PERFORMANCE OF ECC AND COMPARAISON

The inverse operation of ECC which known as the Elliptic Curve Discrete Logarithm Problem (ECDLP) gets harder, faster, against increasing key length than do the inverse operations in Diffie-Hellman and RSA. As security requirements become more stringent, and as processing power gets cheaper and more available, ECC becomes the more practical system for use. And as security requirements become more demanding, and processors become more powerful. This keeps ECC implementations smaller and more efficient than other implementations.

With increasing key length, the inverse operation of ECC, known as the Elliptic Curve Discrete Logarithm Problem (ECDLP), becomes harder and quicker than the inverse operations of Diffie-Hellman and RSA. ECC is becoming the more realistic approach to utilise as security needs get more demanding and computing power becomes cheaper and more available. Security standards are becoming more stringent, while processors are becoming more powerful. As a result, ECC implementations are smaller and more efficient than other approaches. ECC can employ a lot smaller key and still provide the same level of security as other asymmetric algorithms that need much bigger keys. Furthermore, at higher degrees of security, the disparity between ECC and its rivals in terms of key size required for a given level of security becomes considerably more obvious.

The same degree of security, data quantities, encrypted message sizes, and processing power are all compared between the two asymmetric cryptographic methods, RSA and ECC. However, compared to other cryptographic techniques, ECC has smaller keys (RSA).

Table 1.1 : Comparative analysis

ECC Key Size (bits)	RSA Key Size (bits)	Key Size ratio	AES Key Size (bits)
160	1024	1:06	NA
256	3072	1:12	128
384	7680	1:20	192
512	15360	1:30	256

V. CONCLUSION

For security, cryptography methods are commonly utilised in sensor nodes and IoT systems. The limited resources of the nodes' underlying IoT devices, on the other hand, need careful selection of security methods. Light-weight cryptographic methods have been developed particularly for low-power and resource-constrained embedded devices with this in mind. However, depending on the performance and processing needs, one algorithm may be best suited for one application while being unsuitable for another.

VI. REFERENCES

[1] S. R. Moosavi et al., "SEA: A secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways," Procedia Comput. Sci., vol. 52, no. 1, pp. 452–459, 2015, doi: 10.1016/j.procs.2015.05.013.

[2] M. A. Rashid and H. H. Pajooh, "A security framework for iot authentication and authorization based on blockchain technology," Proc. - 2019 18th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. IEEE Int. Conf. Big Data Sci. Eng. Trust. 2019, pp. 264–271, 2019, doi: 10.1109/TrustCom/BigDataSE.2019.00043.

[3] A. Fayad, B. Hammi, and R. Khatoun, "An adaptive authentication and authorization scheme for IoT's gateways: A blockchain based approach," 2018 3rd Int. Conf. Secur. Smart Cities, Ind. Control Syst. Commun. SSIC 2018 - Proc., pp. 1–7, 2018, doi: 10.1109/SSIC.2018.8556668.

[4] T. K. Goyal, V. Sahula, and D. Kumawat, "Energy Efficient Lightweight Cryptography Algorithms for IoT Devices," IETE J. Res., vol. 0, no. 0, pp. 1–14, 2019, doi: 10.1080/03772063.2019.1670103.

[5] A. Lohachab, "Journal of Information Security and Applications ECC based inter-device authentication and authorization scheme using MQTT for IoT networks," J. Inf. Secur. Appl., vol. 46, pp. 1–12, 2019, doi: 10.1016/j.jisa.2019.02.005.

[6] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad, "Identity Establishment and Capability Based Access Control (IECAC) Scheme for Internet of Things," 2012.

[7] M. Tahir, M. Sardaraz, S. Muhammad, and M. S. Khan, "A lightweight authentication and authorization framework for blockchain-enabled IoT network in health-informatics," Sustain., vol. 12, no. 17, 2020, doi: 10.3390/SU12176960.

[8] V. Beltran and A. F. Skarmeta, "An Overview on Delegated Authorization for CoAP," pp. 706–710, 2016.

[9] N. Saxena, B. J. Choi, R. Lu, and S. Member, "Authentication and Authorization Scheme for Various User-Roles and Devices in Smart Grid," vol. XX, no. X, 2015, doi: 10.1109/TIFS.2015.2512525.

[10] W. Ren, "Lightweight and Robust Schemes for Privacy Protection in Key Personal IoT Applications : Mobile WBSN and Participatory Sensing," 2016.

[11] V. Venukumar and V. Pathari, "Multi-Factor Authentication Using Threshold Cryptography," pp. 1694–1698, 2016.

[12] R. Kolisch and S. Hartmann, "HEURISTIC ALGORITHMS FOR THE RESOURCE-CONSTRAINED PROJECT SCHEDULING PROBLEM :," 1999.

[13] O. G. B, R. Rietman, S. Sharma, L. Tolhuizen, and J. L. Torre-arce, "A Comprehensive and Lightweight Security Architecture to Secure the IoT Throughout the Lifecycle of a Device Based on HIMMO," pp. 112–128, 2018, doi: 10.1007/978-3-319-28472-9.

[14] G. A. Lewis and D. Klinedinst, "Authentication and Authorization for IoT Devices in Disadvantaged Environments," pp. 368–373, 2019.

[15] C. Science, R. Kumar, C. Science, E. Vvce, and C. Science, "Comparative Study of Various Lightweight Between IoT and Cloud," no. Icces, pp. 589–593, 2020.

[16] M. B. B. A. Malar and J. Prabhu, "Trust based authentication scheme (tbas) for cloud computing environment with Kerberos protocol using distributed controller and prevention attack," 2020, doi: 10.1108/IJPCC-03-2020-0009.

[17] X. Iru et al., "A Mutual Authentication Protocol for IoT Devices," pp. 716–720, 2018.

[18] S. Kallam, "Diffe-Hellman: Key Exchange and public key cryptosystems," 2015.

[19] Z. Hu, Y. Zhu, and L. Ma, "An improved Kerberos protocol based on Diffie-Hellman-DSA key exchange," IEEE Int. Conf. Networks, ICON, pp. 400–404, 2012, doi: 10.1109/ICON.2012.6506591.

[20] M. Amara and A. Siad, "Elliptic Curve Cryptography and its applications," 7th Int. Work. Syst. Signal Process. their Appl. WoSSPA 2011, pp. 247–250, 2011, doi: 10.1109/WOSSPA.2011.5931464.

[21] N. Samir et al., "ASIC and FPGA Comparative Study for IoT Lightweight Hardware Security Algorithms," J. Circuits, Syst. Comput., vol. 28, no. 12, 2019, doi: 10.1142/S0218126619300095.

[22] V. K. Sarker, T. N. Gia, H. Tenhunen, and T. Westerlund, "Lightweight Security Algorithms for Resource-constrained IoT-based Sensor Nodes," IEEE Int. Conf. Commun., vol. 2020-June, 2020, doi: 10.1109/ICC40277.2020.9149359.

[23] M. O. D. Evices, "a L Ightweight R Econfigurable S Ecurity M Echanism for J Alal a L -M Uhtadi , D Ennis M Ickunas , and R Oy C Ampbell ," Ieee Wirel. Commun., no. April, 2002.

[24] C. Wang and C. Feng, "Security analysis and improvement for kerberos based on dynamic password and diffie-hellman algorithm," Proc. - 4th Int. Conf. Emerg. Intell. Data Web Technol. EIDWT 2013, vol. 1, no. 1, pp. 256–260, 2013, doi: 10.1109/EIDWT.2013.49.

[25] B. Zhou, M. Egele, and A. Joshi, "High-performance low-energy implementation of cryptographic algorithms on a programmable SoC for IoT devices," 2017 IEEE High Perform. Extrem. Comput. Conf. HPEC 2017, 2017, doi: 10.1109/HPEC.2017.8091062.

VII. AUTHORS PROFILE

Ansari Maqsood Ahmed Research Scholar in Computer Engineering Dept. in Shri. JJT University, Jhunjhunu Rajasthan. Having 22+ years of experience in the field of Technical Education. During this tenure, I authored 178 technical books for various universities across India. , I also published 10 technical papers in reputed journals and conferences. Also having life t time membership of ISTE. Guest Faculty for BITS WIPRO collaboration -WASE Programme at WIPRO Pune.

Dr. Prasadu Peddi Currently working as Assistant Professor in Shri. JJT University, Rajasthan having 4+ years of experience in IT industry and 8+ years of experience in Teaching. Reviewer for a web of science and springer series journal, Member in IEEE and International Association of Engineers Specialization in Natural language Processing, Speech Processing, Image Processing, Network & Web Security ,Grid Computing. Guest Faculty for BITS WIPRO collaboration -WASE Programme at WIPRO Hyderabad.