2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST) | 978-1-6654-5866-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCMST54943.2021.00021

Cryptographic algorithms in IoT- a detailed analysis

Pavandeep Kaur
Apex Institute of Technology
Chandigarh University
Punjab, India
pavubansal23@gmail.com

Shivani Aggarwal
Apex Institute of Technology
Chandigarh University
Punjab, India
aggarwal.er@gmail.com

***Abstract*— IOT is quickly becoming a fundamental platform for interconnected devices. With such an advancement in technology, interconnected devices are also having security issues. As a matter of fact, Internet of Things (IoT) security has become a critical challenge. Because scalable systems and services are subject to a variety of vulnerabilities and data breaches, more confidentiality and anonymity are necessary. This paper presents cryptographic approach using symmetric and asymmetric encryption techniques for IoT security along with their comparison. This research aims to provide a detailed examination of RSA, Blowfish, Diffie-Hellman, ECC and other cryptographic schemes.**

Keywords*— *Internet of things, cryptography, algorithms, encryption, decryption

1. Introduction

This study examined cryptography research in relation to IoT security in depth. Kevin Ashton coined the phrase "Internet of Things" in 1998 to characterize the future of the internet and ubiquitous computing [1]. "Internet of Things" is abbreviated as "IoT." The Internet of Things (IoT) is a built-in innovation that allows "Things" to be physically connected and accessible through the internet. In this sense, "things" can refer to anything that can interact with one another without the need for human involvement, such as household appliances, vehicles, machinery, and so on. Machine-to-machine communication [2] is a means of linking smart devices.

In the IoT ecosystem, security concerns like as privacy, safe storage and administration, authorization, communication, and access control are critical and complex challenges. The widespread adoption of IoT sensors and applications results in a plethora of network security flaws and attacks. Common security models suffer from a variety of drawbacks as a result of their restricted processing capabilities. As a result, IoT security should be increased by ensuring connectivity, allowing only authorized users to access data. Cryptography algorithms are the most secure methods for IoT resources.

The security of IoT applications in the workplace is another big potential threat. When an unauthorized entrant obtains the sensing data from a smart device from a major corporate unit, the attacker use it to spy on the company. As a result, It is possible that IoT security will be extra difficult to achieve than traditional security[3]. This is because the Internet of Things (IoT) includes a collection of networks that require security in a range of domains, including the Internet, sensor networks, as well as mobile networks. Additional challenges like as administration, access control, authentication, and privacy occur when many networks are joined in this way [4]. This is when cryptography enters the picture.

ORGANIZATION OF THE PAPER

Rest of the paper is organized as – Section 2 describes cryptography, Section 3 consists of different cryptographic algorithms. Section 4 depicts the performance comparison of cryptographic algorithms. Section 5 contains the results. Section 6 concludes the paper.

2. Overview of Cryptography

The application of mathematical techniques to offer security services such as secrecy, data integrity, authentication, and authorization is known as “cryptography” [5]. As seen in Figure 1, cryptography is the solution to IOT security requirements.

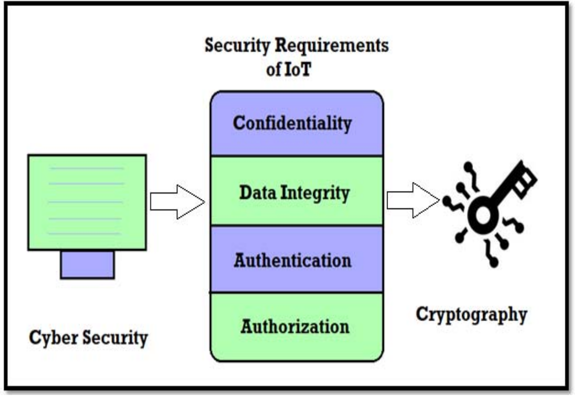


Fig.1. Cryptography as a security service

Cryptography is the study of transforming plaintext into an unreadable form (cipher-text) by encryption and then decrypting the cipher-text back to plaintext. Cryptography is a new term that secures data more efficiently while using less resources and delivering higher throughput, stability, and low power consumption. The cryptographic algorithms are further classified into 2 types: (a) Symmetric and (b) Asymmetric.

Any step done to prevent a computer system against any unauthorized access or information exploitation by an intruder is referred to as “security”. On a daily basis, hostile activities include the breach of confidentiality, credit card robberies, the leaking of economic information, and malware infection on computer systems [6].

The following recommendations are made in order to achieve computer network protection:

- **Confidentiality**: It protects system information from illegal access coming from external sources. It is maintained in a variety of approaches. The most prevalent methods for accomplishing this are cryptographic techniques and access control. Cryptography is the technology used to encode the

978-1-6654-5866-5/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCMST54943.2021.00021

target data, and it cannot be employed by any party unless it has the decryption key. RSA, DSA, and AES are examples of popular encryption algorithms. Inappropriate accessibility to confidential material is prohibited to anybody whose actual identity cannot be confirmed throughout the access control procedure. [7].

- **Integrity**: data that have not been altered with by an unintelligible entity are said to be integrated [8].
- **Authentication**: Authentication is the process of confirming a system resource user's identity [8].
- **Non-repudiation**: A node can never deny an action it has taken, such as sending a message.
- **Availability**: Authorized people have accessibility to system resources.
- **Confidentiality**: the user's identity cannot be determined from his actions in the system.
- **Authorization:** It is a means of determining user permissions or access levels to system resources.

3. DIFFERENT CRYPTOGRAPHIC ALGORITHMS

Message integrity checks, encryption, entity authentication, and other services are given by symmetric ciphers, but asymmetric ciphers also provide non-repudiation and key management. Symmetric ciphers are classified into 2 types - stream cipher and block cipher.

(a) Stream ciphers create an output cipher stream by synthesizing plain-text digits with a pseudo-random key stream. They are also known as “state ciphers” since the pseudo-random secret sequence is produced via a hidden internal state and these ciphers process the message a bit or byte at a time. The key stream is frequently merged with the plain-text using a bit-wise “XOR” operator during the encryption process. The identical key stream is generated at the receiver end, and it is used to decode the cipher and return the plaintext.

(b) Block Ciphers are encryption algorithms that process data in blocks of fixed sizes. To produce cipher-text blocks, plaintext blocks get combined with a key. It is used for encrypting the data, message integrity & authentication, random bit formation, hashing, and other similar tasks. The most significant block ciphers, presumably, are AES and DES.

Figure 2 depicts symmetric and asymmetric cryptographic algorithms.

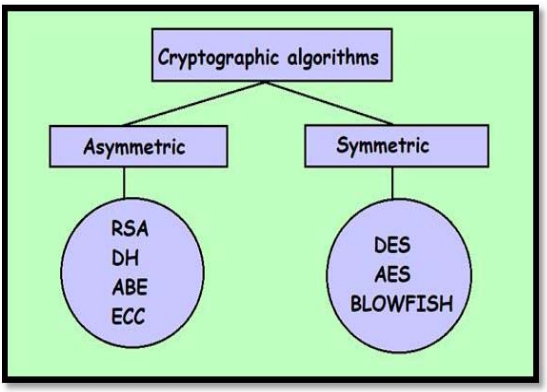


Fig.2. Cryptographic algorithms

3.1 ASYMMETRIC ALGORITHMS

Asymmetric cryptography uses two unique keys, it provides more security: a public key that is only used to encrypt data, keeping it secure for everyone to use, and a private key that is only used to decode messages yet don't ever needed to be shared.

3.1.1 RSA

RSA being a public-key cryptography technique that provides encryption and digital signatures [9] was called after its developers (Rivest, Shamir, and Adleman) in 1977. It is indeed modeled on the computational challenges of factoring large prime numbers. To encrypt and decode data, a public key (e) and a private key (d), both positive integers, are employed. Here's how it works [18]:

Step 1. Select any 2 random values p,q (p must not be equal to q, and both should be prime). Let p=17, q=11.

Step 2. Calculate n as:

$$n = p \text{ x } q = 17 \text{ x } 11 = 187$$

Step 3. Next Φ(n) is calculated by:

$$\Phi(n) = (p-1)(q-1) = (17-1)(11-1) = (16)(10) = 160$$

Now take value of public key ‘e’ such that e is relatively prime with Φ [range 1< e < Φ(n) i.e 1< e < 160]. Let e=7.

Step 4. Now calculate private key ‘d’ as:

$$d = e\text{^}(-1) \bmod [\Phi(n)] = 7^{-1} \bmod 160 = 23$$

Manually put values, (n,e)=(187,7) and (187,23)= (7,23)

For example, if the message value is 4, then ciphertext can be calculated as:

$C = m^{e} \bmod n = C = 4^{7} \bmod 187 = 16384 \bmod 187 = 115$

Therefore, the encrypted message (*c*) is 115.

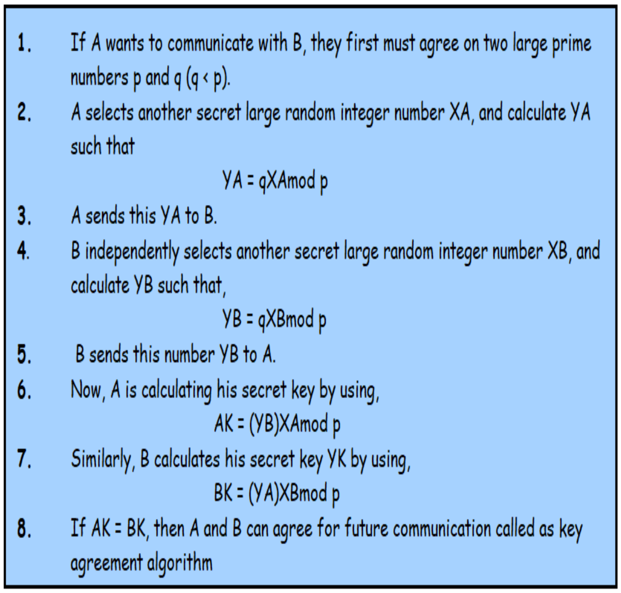
The encrypted message (*c*) is then decrypted with:

$M = c^{d} \bmod n = 115^{23} \bmod 187 = 4$, which is equal to the message value.

RSA is more secure than any other symmetric key technique, and its benefits in cryptography include authenticity and confidentiality. The sole disadvantage is that RSA requires excessive processing.

3.1.2 DH (Diffie Hellman)

(Whitfield Diffie and Martin Hellman, 1976) developed the Diffie-Hellman (DH) algorithm, which is a cryptographic technique [10]. It lets 2 entities to communicate by agreeing on a symmetric key, which is used to encode and decode data. The algorithm is created using mathematical ideas. DH is an algorithm for sending a shared secret among two users via a public network. This shared secret is required in order for two users who have never communicated before to encrypt their communications. Figure 3 shows steps of Diffie Hellman algorithm.



1. If A wants to communicate with B, they first must agree on two large prime numbers p and q (q < p).
2. A selects another secret large random integer number XA, and calculate YA such that

   YA = qXAmod p
3. A sends this YA to B.
4. B independently selects another secret large random integer number XB, and calculate YB such that,

   YB = qXBmod p
5. B sends this number YB to A.
6. Now, A is calculating his secret key by using,

   AK = (YB)XAmod p
7. Similarly, B calculates his secret key YK by using,

   BK = (YA)XBmod p
8. If AK = BK, then A and B can agree for future communication called as key agreement algorithm

Fig.3. Steps of DH algorithm

3.1.3 ABE

ABE is an abbreviation for “attribute-based encryption”, which is a sort of public key encryption. In this cryptographic technology, the encryption of data is done using a Boolean formula. This formula is called access policy, which other parties must fulfil in attempt to decipher the cipher text [12]. This approach is highly helpful on the Internet of Things since it offers fine access control as well as encryption.

The two basic types of ABE approach are (a) key policy (KP-ABE) and (b) cipher-text policy (CP-ABE).

(a) **KP-ABE** - (Goyal et al.) developed the first KP-ABE implementation [24], which enabled access policies to somehow be defined using any monotonic formula over encrypted data. Under the Bilinear Diffie-Hellman assumption, the system was shown to be selectively secure. Later, by including revocation methods into that KP-ABE scheme, Ostrovsky et al. [25] introduced a KP-ABE system in which private keys can reflect any access formula over attributes, including non-monotonic ones.

(b) **CP-ABE** – It combines the tree access structure into cipher-text and construct the users' secret keys by mixing attribute sets [19]. The CP-ABE algorithm is not the same as the standard ABE method. The quantity of system characteristics has no effect on the length of public keys and parameters. It uses two-level random masks for eliminating the chances of user collision.

Setup, Encrypt, KeyGen, and Decrypt are the 4 core algorithms of a ciphertext-policy attribute-based encryption method. The **setup** algorithm takes implicit security parameter. It returns the public parameters PK and a master key MK as an output. The **Encrypt** method is similar to the **KeyGen** algorithm of KP-ABE, with the exception that Pr(0) = s. The **decrypt** algorithm is identical to KP-ABE, except the number of bilinear pairing operations is twice. Because the access tree is embedded in the cipher-text, data access control is feasible in CP-ABE, however KP-ABE seems to have no access control because the access tree is comprised of the users' key.

3.1.4 ECC

Victor S. Miller and Neal Koblitz introduced ECC in 1985 [16]. ECC's ability to operate on finite domains is a critical feature. Eq. (1) defines elliptic curve as:

$y^2 = x^3 + ax + b$... (1)

ECC is an encryption technique focused primarily on the algebraic mathematical structure that has a smaller message size and requires minimal keys than other public key systems. Even though ECC works with fewer keys but still provides the high degree of sustainability and security.

ECC is always symmetric to x-axis [17]. As shown in figure 4, line touches maximum three points.

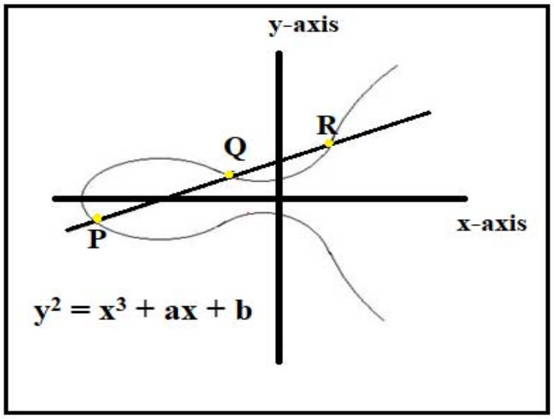


Fig.4. Elliptic curve

Step 1- Let Ep(a,b) be the elliptic curve. Consider the Eq. (2):

$Q=k*P$ ………………………………….. (2)

here we can calculate Q very easily if we know the values of k and P, but it is very difficult to find k if we know the values of Q and P. This problem is called discrete logarithm problem.

Step 2 - Let 'q' be any prime number or an integer of the form 2^m. Let 'G' be a point on the elliptic curve whose order is large value of n.

Step 3 - For user A, Select private key 'n_a' such that $n_a<n$. Calculate public key P_a as: $P_a = n_a$ x G

Step 4 - For user B, Select private key 'n_b' such that $n_b<n$. Calculate public key P_b as: $P_b = n_b$ x G

Step 5 – Calculate secret key by user A, $k_a=k=n_a$ x P_b

Step 6 – Calculate secret key by user B, $k=n_b$ x P_a

Step 7 – Perform encryption as: $C_m=\{kG, P_m + kP_b\}$

To safeguard internal communications, the US government employs the ECC algorithm. It is the technique for proving bitcoin ownership. It adds digital signatures to Apple's iMessage service.

3.2 SYMMETRIC ALGORITHMS

Symmetric algorithms are cryptographic algorithms that employ the similar cryptographic keys for encryption of plain-text and decryption of cipher-text.

3.2.1 DES

The Data Encryption Standard is a symmetric key technique for encrypting data in order to protect it from an attacker or an unauthorized user [13]. DES has a great deal of power in the security field since it protects data. In 1997, the National Bureau of Standards had adopted DES, which is now known as Federal Information Processing Standards. DES encrypts data in 64-bit blocks with a 56-bit key. The method turns 64-bit input into 64-bit output in a succession of stages, as shown in Figure 5. To reverse the encryption, the same techniques and key are required [23].

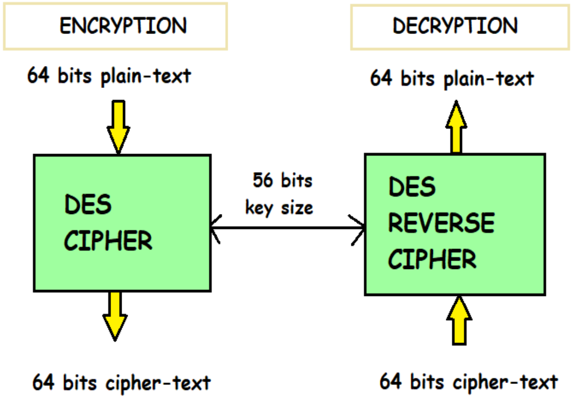


Fig. 5. DES Encryption and decryption process

The Triple DES technique offers more security and requires minimal time to perform operations like encryption and decryption than other cryptography methods. But DES is not secure enough because of too short key length i.e 56 bits.

3.2.2 AES

The AES algorithm encrypts and decrypts data using a single key [14]. The AES algorithm takes input blocks of 128, 192, and 256 bits in size. It is determined by the length of the key.

TABLE I. No. of iterations as per key size

KEY SIZE	NO. OF ITERATIONS
128 bits	10
192 bits	12
256 bits	14

Plain text is turned into encrypted text in this approach after going through many steps as shown in figure 6, such as byte substitution, row shift, mix column, and round key [20].

- **Sub Byte**: For sub-byte round, all bits in state are replaced by the other, as done in Rijndeal S-Box.
- **Shift Row**: All rows are moved to the left by a four - fold.
- **Mix Column**: In this case, the linear transformation is performed on the array's columns.
- **Add Round Key**: After iterations, each byte of the state is coupled with a round key, that is obtained from the Rijndeal key scheme [22].

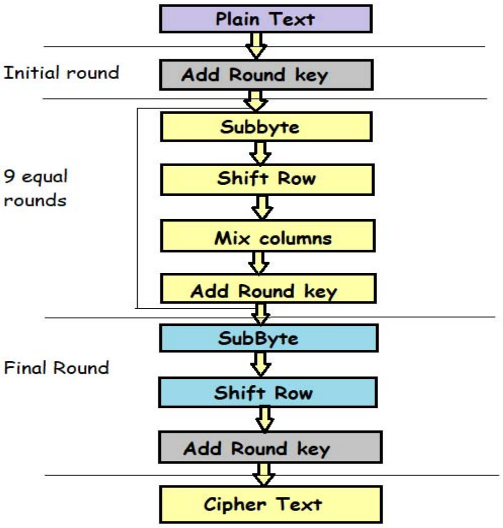


Fig. 6. Working of AES algorithm

3.2.3 BLOWFISH

In 1993, Bruce Schneier invented the blowfish block cypher [15]. It employs a fixed 64-bit block with key lengths varying from 32 to 448 bits. Depending on the switch, it also employs huge S-boxes. It is an adaptable algorithm that has not been cracked. It is also among the fastest ciphers for public usage. Blowfish Algorithm is a Feistel network-based symmetric block cipher that iterates basic cryptographic operations like encryption and decryption 16 times. Feistel is a mechanism for quickly converting any function into a permutation. The sole prerequisite for decrypting the encrypted text is to reverse the key schedule. The crucial point BA expansion begins with the P-array and Sboxes [21] and involves the usage of multiple sub-keys, which necessitates pre-compilation prior to data encryption or decryption.

Steps of Blowfish Algorithm are:

Step 1 - The Blowfish method employs a 64-bit block size, and length of produced key ranges between 32 and 448 bits. The algorithm is divided into two sections. The first is for key expansion, while the second is for data encryption.

Step 2 - Once the request is received, the key expansion turns the 448 bits of a key into sub-keys, causing the array to grow to 4168 bytes in size.

Step 3 - For data encryption, the technique employs a 16-round Feistel cipher along with massive key-dependent S-boxes.

Step 4 - Each cycle of substitution in the S-boxes has a unique permutation key.

In Blowfish, the P-array has 18 entries, whereas the S-boxes contains four 256-entry entries. S-boxes are then employed to transform the 8-bit input to a 32-bit output. Once all the rounds are completed except last one, each half of the data block is XORed along with one of the P-entries that has not yet been utilised. The new adjusted subkeys are then used to encrypt P1 and P2. For producing fresh subkeys for the P-array and the four S-boxes, the Blowfish cipher repeats this method 521 times.

4. COMPARISON OF CRYPTOGRAPHIC ALGORITHMS

TABLE 2. Comparison of cryptographic algorithms

Algorithm	Created By	Key size	Block size	Security	Speed
RSA	Rivest, Shamir, Adleman	1024 to 4096 bits	128 bits	Excellent	Slow
DH	Whitefield Diffie, Martin Hellman	Variable	-	Good	Slow
ABE	Amit Sahai, Brent Waters, Vipul Goyal, Omkant Pandey	-	-	Good	Slow
ECC	Victor S. Miller, Neal Koblitz	Variable	Variable	Excellent	Fast
DES	IBM	56 bits	64 bits	Not secure	Slow
AES	Vincent Rijmen, Joan Daemen	128, 192, 256 bits	128 bits	Adequately secured	Fast
Blowfish	Bruce Schneier	32-448 bits	64 bits	Secure enough	Fast

Figure 7 demonstrates the usage of cryptographic algorithms in IoT.

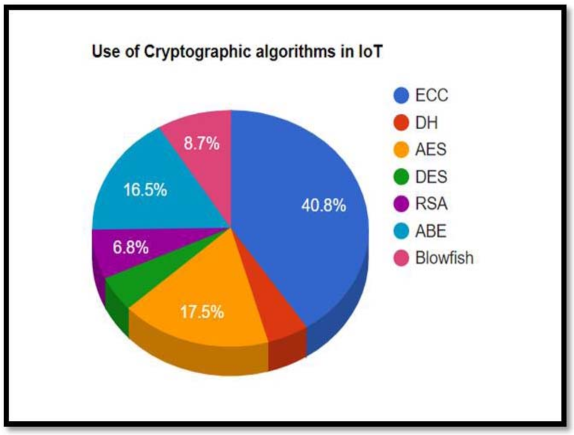


Fig.7. Use of Cryptographic algorithms in IoT

5. RESULTS

ECC is employed to provide excellent security in small devices. In comparison to alternative asymmetric approaches now in usage, ECC employs a minimal amount of keys while providing good security. ECC with a small key size saves money in considerations of memory and computational power. As a result, ECC is strongly advised for the creation of lighter and faster cryptographic algorithms that can run on small machines. RSA also provides excellent level of security but it is significantly slower than ECC. Blowfish is substantially quicker than DES, but the performance improvement is slower since it requires significantly more memory for sub-key and S-box setup. DES is least secure and slow among all other cryptographic algorithms.

6. CONCLUSION

IoT has revealed a major security flaw that affects everything from authentication, authorization to trust management, as well as a danger to its embedding systems. Cryptographic techniques were used to explore IoT security in this study. Security has to be a primary issue while designing the IoT ecosystem. Cryptography algorithms are a highly strong mechanism for safeguarding the network's physical layer, and they are critical for the effective security of the core network design. ECC has shown to be the safest and most efficient encryption method.

REFERENCES

[1] Wu, M. et. al., (2012) "Research on the architecture of Internet of things". The Proceedings of 3rd International Conference on Advanced Computer Theory and Engineering, 20–22 Aug, Beijing, China.

[2] Shah A., Engineer M. (2019) "A Survey of Lightweight Cryptographic Algorithms for IoT-Based Applications". Smart Innovations in Communication and Computational Sciences. Advances in Intelligent Systems and Computing, vol 851. Springer.

[3] Radoglou Grammatikis, P. I., Sarigiannidis, P. G., & Moscholios, I. D. (2019). "Securing the internet of things: Challenges, threats and solutions". Internet of Things, 5, 41–70.

[4] Kouicem, D. E., Bouabdallah, A., & Lakhlef, H. (2018). "Internet of things security: A top-down survey". Computer Networks, 141, 199–221.

[5] Zeadally, S., Das, A. K., & Sklavos, N. (2019). "Cryptographic technologies and protocol standards for internet of things". Internet of Things.

[6] Rauscher, J., & Bauer, B. (2018). "Safety and security architecture analyses framework for the internet of things of medical devices". 2018 IEEE 20th international conference on e-health networking, applications and services.

[7] Lu, X., Pan, Z., & Xian, H. (2019). "An integrity verification scheme of cloud storage for internet-of-things mobile terminal devices". Computers & Security.

[8] Rivest, R., Shamir, A., & Adleman, L. (1978). "A method for obtaining digital signatures and public-key cryptosystems". Communications of the ACM, 21(2), 120–126.

[9] Kandhoul, N., & Dhurandher, S. K. (2019). "An asymmetric RSA-based security approach for opportunistic IoT". 2nd international conference on wireless intelligent and distributed environment for communication. Springer International Publishing.

[10] A. El Emine Sejad, K. Wane Keita, K. Tall and I. Diop, (2020) "Proposal of a DH optimization model," 2020 International Conference on Computer, Information and Telecommunication Systems, pp. 1-5.

[11] Shah, R. H., & Salapurkar, D. P. (2017). "A multifactor authentication system using secret splitting in the perspective of cloud of things". 2017 international conference on emerging trends & innovation in ICT.

[12] Pace, G. J., Picazo-Sanchez, P., & Schneider, G. (2018). "Migrating monitors ? ABE: A suitable combination for secure IoT?" Leveraging applications of formal methods, verification and validation. Industrial practice. Springer International Publishing.

[13] Chandi, P., Sharma, A., Chhabra, A., & Gupta, P. (2019). "A DES-based mechanism to secure personal data on the internet of things". In ICCCE 2018. Springer.

[14] Cruz-Duarte, S., Sastoque-Mahecha, M., Gaona-Garcı´a, E., & Gaona-Garcı´a, P. (2019). "Security scheme for IoT environments in smart grids". In Information systems and technologies to support learning. Springer International Publishing.

[15] Schneier, B. (1993). "Description of a new variable-length key, 64-bit block cipher (blowfish), fast software encryption". Cambridge security workshop proceedings. Springer (pp. 191–204).

[16] Miller, V. S. (1986). “Use of elliptic curves in cryptography”. Advances in cryptology—CRYPTO ’85 proceedings. Berlin, Heidelberg: Springer.

[17] Mousavi, S.K., Ghaffari, A., Besharat, S. *et al.* (2021) “Security of internet of things based on cryptographic algorithms: a survey”. *Wireless Netw* **27,** 1515–1555.

[18] Xin Zhou and Xiaofei Tang, (2011) "Research and implementation of RSA algorithm for encryption and decryption," Proceedings of 2011 6th International Forum on Strategic Technology, pp. 1118-1121.

[19] Y. Yan, M. B. M. Kamel and P. Ligeti, (2020) "Attribute-based Encryption in Cloud Computing Environment," 2020 International Conference on Computing, Electronics & Communications Engineering, pp. 63-68.

[20] S. Mewada, P. Sharma and S. S. Gautam, (2016) "Exploration of efficient symmetric AES algorithm," 2016 Symposium on Colossal Data Analysis and Networking, pp. 1-5.

[21] A. Alabaichi, F. Ahmad and R. Mahmod, (2013) "Security analysis of blowfish algorithm," 2013 Second International Conference on Informatics & Applications (ICIA), pp. 12-18.

[22] B. Bhat, A. W. Ali and A. Gupta, (2015) "DES and AES performance evaluation," International Conference on Computing, Communication & Automation, pp. 887-890.

[23] O. Reyad, H. M. Mansour, M. Heshmat and E. A. Zanaty, (2021) "Key-Based Enhancement of Data Encryption Standard For Text Security," 2021 National Computing Colleges Conference (NCCC), pp. 1-6.

[24] V. Goyal, O. Pandey, A. Sahai, and B. Waters, (2006) “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 89–98..

[25] R. Ostrovsky, A. Sahai, and B. Waters, (2007) “Attribute-based encryption with non-monotonic access structures,” in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 195–203.

Author(s) Profile

Pavandeep Kaur has received her Bachelor in Computer Science Engineering (Internet of Things) from Chandigarh University, Punjab. She is currently pursuing her Masters in CSE (Cloud Computing) in collaboration with Virtusa from Chandigarh University, Punjab. Her key areas of interest include Internet of Things and Cloud Computing.

Shivani Aggarwal is an Assistant Professor at AIT-CSE department in Chandigarh University, Punjab. Her area of expertise includes Artificial intelligence, Machine learning, Soft Computing, and Deep learning.