International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Handling Information Security Wisely Utilising the Aggressive Cuckoo Search Algorithm in Lock Systems

Adusumalli Balaji[1*], Ch. Indira Priyadarsini[2], Eluri Nageswara Rao[3], Gowru Bharath Kumar[4], Arumalla Nagaraju[5], Narne Srikanth[6], Popuri Srinivasarao[7]

Submitted: 15/12/2023 **Revised**: 20/01/2024 **Accepted**: 03/02/2024

Abstract: Concerns relating to safety and privacy have been brought to light as a result of the growing adoption of Internet of Things (IoT) devices, such as smart home appliances, mobile phones, and smart watches. Security has emerged as an essential component in the development of Internet of Things (IoT) systems in light of the growing number of assaults that originate from both malicious and non-malicious sources. The increasing volume of data that is stored in IoT systems presents a number of unique challenges, one of the most crucial being data security. Hackers may be able to remotely take control of Internet of Things devices if inadequate security measures are in place, which may result in substantial harm. This research suggests utilizing the Aggressive Cuckoo Search Algorithm method (ACS), the SHA-256 method, and the Elliptic Curve Cryptography (ECC) in order to solve these issues while enhancing the safety of internet of things (I smart door lock systems. ACS stands for Aggressive Cuckoo Search, SHA-256 stands for Secure Hash Algorithm, and ECC stands for Elliptic Curve Cryptography. In order to produce the ECC private key, the suggested architecture makes use of ACS, which has the potential to improve the data storage security of IoT systems. In the study, both the encoding and decoding times of the suggested design are analyzed, and it is compared to other encoding methods like (ECC-GA-SHA-256) and (ECC-FA-SHA-256). The findings indicate that the proposed design is capable of producing superior outcomes with 125 iterations for the encoding process and 175 iterations for the decoding process. In addition, the proposed design features encoding and decoding times that are over 16.17 percentage points faster than those of previous encoding methods. Based on these data, it appears that the design that was proposed has the potential to considerably improve the functionality and safety of IoT-based smart door lock systems. The employment of ACS, SHA-256, and ECC can serve as potential solutions to overcome the security concerns connected with IoT-based intelligent locking systems. These are all cryptographic standard algorithms.

Keywords: *Information security; IoT; cryptography; Aggressive Cuckoo Search Algorithms*

1. Introduction

The Internet of Things (IoT) is a technical invention that has resulted in a huge industry-wide shift, particularly in the areas of communication and information. The technology functions thanks to the collaboration of intelligent servers, workstations, and sensor equipment, which makes it possible to incorporate a wide variety of items that can detect certain things [1]. The Internet of Things can be used for a wide variety of things, including consumer electronics like smartphones and home appliances.

*[*1]Computer Science and Engineering, Chalapathi institute of engineering and technology, Guntur, India.*
[2]Deapartment of Mechanical Engineering, Chaitanya Bharati Institute of Technology, Hyderabad, India.
[3]Computer Science and Engineering(IOT), RVR & JC College of Engineering, Guntur, India.
[4]Computer Science and Engineering, Chalapathi institute of engineering and technology, Guntur,India
[5]Computer Science and Engineering, koneru lakshmaiah education foundation, vaddeswaram,Vijayawada,India
[6]Computer Science and Engineering(AI&ML), RVR & JC College of Engineering, Guntur, India [7]
[7]Computer Science and Engineering(DS), RVR & JC College of Engineering, Guntur, India.
Email:srinivas.popuri4@gmail.com

[2,3] Everything from smartwatches to healthcare systems to governmental services. Nevertheless, the Internet of Things (IoT) has been met with a number of obstacles in the way of security, particularly in the realms of authentication, confidentiality, privacy, and integrity [4]. Because of the enormous amount of data that is produced by IoT, securing data security has become an extremely important concern. A network's data can be put at risk of security breaches and suffer irreparable harm if it is replicated or modified without authorized access [5,6]. Therefore, in order to protect the data generated by IoT devices, robust security mechanisms are required [7]. These robust security mechanisms should include network authentication protocols, key management methods, and security measures between IoT gates and wireless sensors.

The installation of a smart door lock system is an important use of the Internet of Things. It is of the utmost importance to make certain that the data stored in such systems are protected, and protective measures including anti-tampering and data integrity must be put into place in order to guarantee the safety of communications [8,9]. Cryptography continues to be the most effective method for protecting data [10], despite the fact that the small size of IoT sensing devices reduces the likelihood of hardware-related risks. It is essential for Internet of Things-based smart door lock systems to incorporate cryptographic algorithms since these algorithms have the potential to optimize and automate security measures [11].

A hybrid model has been suggested as a means of improving the data security offered by the IoT. This model proposes a combination strategy that makes use of SHA-256, ECC, and CS algorithms [12]. A CS-based approach is used to produce the private key for ECC, and the SHA-256 hashing algorithm is used to secure the data after it has been hashed. The actions of cuckoo birds served as the basis for the development of an optimization algorithm known as the CS Algorithm. It does this by iteratively increasing the quality of candidate solutions in the population, exploring new solutions using random walks, and generating new solutions using the cuckoo egg replacement method [12]. This process ultimately results in the generation of the best possible key. The quality of the solutions in the population is improved iteratively by the algorithm, and over the course of time, it converges on the best possible solution.

[13] Electronic Communications Cooperative (ECC) is a sort of public-key and asymmetric encryption technique that secures the secrecy, integrity, and authenticity of data transported via the Internet. The Secure Hash Algorithm 256 (SHA-256) is a popular cryptographic hash function that always gives an output of 256 bits in length, regardless of the size of the data that is being hashed [14]. It protects the integrity of the data by identifying any changes made to the data while it is being transmitted. Even if only one piece of data is altered while being transmitted, the hash value produced by SHA-256 will be distinct from the original hash value, suggesting that the data has been altered [15]. This is an indication that the data has been compromised. A secure method of communication can be maintained over the internet by utilizing ECC for key exchange and digital signatures and SHA-256 for data integrity checks.

1.1. Motivation

It is absolutely necessary to protect the data collected by IoT devices from being accessed by unauthorized parties, and this can be accomplished by utilizing robust encoding techniques throughout the data's storage and transmission. In the internet of things, a breach in data security could present substantial issues in terms of protecting the accuracy and confidentiality of user data. As a result, encrypting the data before it is transmitted is an absolutely necessary step. This research presents a method for achieving efficient data security that makes use of the ACS algorithm to produce a private key for ECC and also incorporates the SHA-256 algorithm. Both of these algorithms are used in tandem.

1.2. Contributions

The findings of this study can be broken down into four categories. To begin, it presents an innovative method for securing IoT data that makes use of ECC, the ACS algorithm, and SHA-256. This method was developed by the authors. Second, by utilizing the ACS algorithm, it generates an effective private key for the ECC. Thirdly, it improves data security by encoding the encrypted text based on ECC-CS with SHA-256. This helps prevent unauthorized access to the data. In conclusion, the viability of the strategy that was suggested is validated by conducting an empirical study in which it is contrasted with several models that are already in use.

1.3. Organization

The following outline constitutes the paper's structure: In Section 2, we will discuss the works that are relevant. In Section 3, you will get an explanation of both the ACS algorithm and the ECC. In Section 4, we will discuss the model that has been proposed. The evaluation and contrast of the suggested model with the other models is shown in Section 5. The report is then finished up with a discussion of potential future research in Section 6.

2. Related Work

In this section, a comprehensive evaluation of previous research that is important to information security in the context of the Internet of Things (IoT) is provided. One study, which was detailed in [16], described a new way for encrypting binary images utilizing a stream cipher and ant colony optimization (ACO) based key generation. This method overcomes the constraints of existing image encoding methods and is therefore superior to those approaches. In addition to this, a comprehensive analysis of the suggested procedure is provided in the paper. In the paper [17], a novel secure visual secret sharing (VSS) method is given. This strategy makes use of both ECC and optimization approaches in order to solve several drawbacks of previous VSS schemes. These disadvantages include the requirement of a high number of shares as well as the potential of unauthorized parties reconstructing shares. In addition, the reference [18] presented a method of image encoding that enhanced the security of picture encoding by utilizing an efficient key generation process based on ECC and the Genetic Algorithm (GA). The purpose of this method was to generate a key that was extremely complex and unpredictable, and one that could withstand attacks such as brute-force and dictionary attacks. A cryptanalysis of the Simplified Data Encoding Standard (SDES) using genetic and memetic algorithms has been performed to find problems in the SDES algorithm. These weaknesses were found by breaking the encoding key and releasing the plaintext message. This information can be found in reference [19]. A wireless network interface selection algorithm for industrial mobile devices has also been proposed in [20]. This algorithm is based on artificial bee colony optimization (ABC) and aims to minimize energy consumption by intelligently selecting the appropriate wireless interface based on the conditions of the network. The algorithm was developed for industrial mobile devices. The authors of [21] proposed a new method of picture encoding that made use of Particle Swarm Optimization (PSO) and a chaotic map to build a key that was extremely randomized and complex. This key was able to withstand multiple types of attacks, including brute-force attacks and dictionary attacks. Additionally, the reference [22] proposed a modified ECC algorithm for authentication and encoding in communication systems. This algorithm used a combination of particle swarm optimization (PSO) and the CS algorithm to optimize the parameters of the ECC algorithm, with the goal of improving the algorithm's efficiency as well as its security. Furthermore, reference [23] proposed a secure data hiding scheme that makes use of a hybridized algorithm that combines the Fruit Fly Optimization (FFO) algorithm and the Seeker Algorithm in order to conceal sensitive data within an image in a way that is both secure and efficient, while also ensuring that the quality of the image is not negatively impacted in any way. This method is known as the Fruit Fly Optimization (FFO) algorithm. A medical picture security strategy that uses a dual encoding approach was also proposed [24]. This approach combines the Advanced Encoding Standard (AES) algorithm with the Rivest-Shamir-Adleman (RSA) algorithm. The optimization of the encoding parameters and the enhancement of the image's level of security were both goals of the suggested method, which utilized an algorithm known as Oppositional Based Optimization (OBO). In [25], the application of the Harmony Search (HS) algorithm in Public Key Cryptography (PKC) was investigated. PKC is a popular form of cryptography that makes it possible to have secure communication across an unsecured network by utilizing public and private keys. The RSA and ElGamal algorithms, which are considered to be the industry standards, were evaluated alongside a revolutionary approach that was introduced in this study. During this time, reference [26] came up with a novel strategy for the encoding of images by combining the signcryption method and the adaptive elephant herding optimization (AEHO) algorithm. Particle swarm optimization (PSO) techniques were utilized in another research study to conduct an analysis of the Geffe generator cryptosystem [27]. For the purpose of producing pseudo-random sequences, the Geffe generator is a stream cipher that is based on linear feedback shift registers and sees widespread usage. In addition, researchers from [28] presented a way to improve the energy efficiency of Cognitive Radio Networks (CRNs) that are dependent on the Internet of Things by applying an algorithm known as multi-objective ant colony optimization (MACO).

In addition to the double-Q learning technique. Both researches revealed novel and potentially useful strategies for improving various facets of network performance. In addition to this, a resource allocation solution for sensor networks in the Internet of Things environment that is based on an improved chaotic firefly algorithm (ICFA) has been proposed in [29]. The method that was suggested was intended to improve the effectiveness of the process of resource allocation by picking the sensor nodes that were the most appropriate for the network on the basis of a number of parameters including energy efficiency, network coverage, and data transmission rate. Furthermore, the publication referred to in footnote 30 provided a novel paradigm for optimizing resource allocation in Time Division numerous Access (TDMA)-based Medium Access Control (MAC) protocols on the Internet of Things. This framework took into account numerous objectives, such as energy efficiency, network performance, and fairness. In the research, a multi-objective optimization technique was utilized in order to discover the optimal method [48]for resource distribution. In [31], another study presented a proposal for a hybrid lightweight encoding and swarm optimization system for the purpose of protecting medical data in applications related to healthcare. The purpose of the suggested method was to ensure efficient use of computational resources and low levels of energy consumption while simultaneously ensuring the safe transmission and storage of medical data. In addition, reference [32] developed an energy-efficient clustering method for the Internet of Things utilizing an Artificial Bee Colony (ABC) optimization approach. The goal of this algorithm was to enhance the lifetime of IoT networks by lowering the amount of energy that individual nodes consumed, all the while guaranteeing that data transmission and cluster formation were carried out effectively. An ideal ECC algorithm was proposed in reference number 33 for the purpose of ensuring the safe storage and transmission of data in big data environments. This approach featured an efficient access control system that restricted illegal access to the data. In a different study, which was summarized in [34], an optimized GA-based strategy was utilized for the purpose of efficient cluster head election in Healthcare Wireless Sensor Networks (HWSNs) that had movable sinks. The approach that was developed intended to increase the network's longevity as well as its energy efficiency by picking the most suitable cluster heads based on criteria such as the amount of energy that was left over, the distance to the sink, and the connectivity to other nodes. In addition, utilizing an addition chain optimization approach that is based on Simplified Swarm Optimization (SSO) and Particle Swarm Optimization (PSO), [35] an improved security scheme for the RSA and ECC algorithms was presented. The RSA and ECC algorithms are frequently utilized in mobile devices for the purpose of ensuring that communications are kept private. The purpose of the suggested system was to make these algorithms more secure. In conclusion, the reference [36] presented a workable answer to the problem of resolving the security concerns that are connected to the IoT systems. These concerns include confidentiality, integrity, and authentication. The proposed method, which makes use of the ABC algorithm, has been tested on a smart irrigation system, and the findings demonstrate that it is effective in securing the data that is transmitted from one device to another. In conclusion, the works reviewed here propose novel strategies for resolving security concerns in a variety of contexts, which, if implemented, have the potential to improve the systems' levels of security and reliability.

3. Preliminaries

This section presents an overview of the ACS and ECC algorithms.

3.1. Aggressive Cuckoo Search Algorithm

In this part of the article, a new aggressive cuckoo search algorithm is suggested. This algorithm is based on some changes that were made to the conventional cuckoo search algorithm.

The aggressive cuckoo is part of a group of bird species known as the Obligate Interspecific Brood Parasites. These birds, which include the aggressive cuckoo, have developed a parasitic behavior as a result of the aggressive cuckoo's quick egg-laying process. This behavior consists of the birds laying their eggs in the nests of other birds and then abandoning the young to develop and hatch on their own without providing any kind of care. As a consequence of this, the breeding success of the host is negatively impacted by a decrease in their nesting success. This can take place in a number of different ways, including the female parasites perhaps destroying or injuring host eggs, or young parasites evicting or killing host young after they have hatched. The aggressive Cuckoo Search [46] technique was designed in order to handle optimization problems that were inspired by nature. This algorithm works by building solutions in nests, which are then improved by altering certain elements of the solution and discarding weaker nests. In order to generate new solutions that are even further streamlined and effective, the ACS algorithm makes use of Lévy flights and Adaptive Step Size, Markov chain random walk.

The French mathematician Paul Pierre Lévy coined the phrase "Lévy flights" in the 1930s. He was the first person to use the term. The term "Lévy flights" refers to a particular kind of random walk in which the durations of individual movement follow the probability distribution of a power law tail. This makes it possible for movements that are uncommon but substantial to take place. This pattern may be seen in a wide variety of natural systems, such as the pattern of foraging activity exhibited by spider monkeys and the flying patterns of certain mammals.

The power law distribution reflects the reality that it may be more beneficial to make a few long-distance journeys sometimes rather than a large number of shorter ones. Lévy flights, although appearing to be an apparently random process, have been discovered to be the optimal answer for optimizing certain search algorithms, which has led to its implementation in a variety of sectors, including biology and finance [37].Lévy [38] discovered that the equation y |x|−α , where α is between 1 and 2, leads to a stable and balanced distribution. Mantegna [39] created an algorithm to simulate this distribution, and the size of each step is determined by a random variable described in Eq. (1).

$$\eta = x/\ |y|1/\beta \qquad (1)$$

The standard deviation of the variable σ x can be determined by applying Eq. (2) when the variables

x and y follow a normal distribution with a mean of 0 and the standard deviation of σ y is 1.

$$\sigma_x = \frac{\Gamma\ 1+\beta\ \sin(\ \frac{\pi\beta}{2})(}{[\Gamma(1\ +\ \beta)/2]\ \beta 2^{(\beta-1)/2}}\ ^{1/\beta} \qquad (2)$$

When the standard deviation of x (x) is 0.6965 and the standard deviation of y (y) is 1, it is recommended that you use 1.5 in equation (2). This advice was made in reference [39], which states that the standard deviation of x should be used. Under these circumstances, when the absolute value of is greater than or equal to 10, the symmetrical Lévy stable process and the probability distributions of exhibit comparable asymptotic behavior. This is the case when is higher than or equal to 10. According to the findings presented in reference [40], which assessed the efficacy of three distinct approaches to the generation of Lévy noise, the Mantegna algorithm proved to be the most efficient of the bunch.

Multiple algorithms, including the CS, ECC, and SHA-256, are incorporated into the data security paradigm that has been presented. While the CS method is responsible for generating private ECC keys, the SHA-256 hashing technique assures both privacy and security. This all-encompassing solution protects the data's confidentiality, integrity, and authenticity, making it suited for a wide range of applications that call for a high level of security. Some examples of these applications include online banking, online shopping, and government computer systems.

3.2. Elliptic-Curve Cryptography

ECC, or elliptic curve cryptography, is a method of encrypting data using mathematical operations performed on very tiny fields. Because of the relatively low length of its keys, this encoding method is significantly faster than others. Because of the difficult exponential discrete logarithm problem [13], the encryption system known as ECC can keep its secrets safe. Two numbers, denoted a and b, drawn from a certain set known as Fp are required in order to perform ECC[47]. In addition, an elliptical curve has a set of points that are labeled as E(Fp), and Q is one of these points, as demonstrated by Equation (3).

$y2 = x3 + ax + b$ (3)

The encryption of data using ECC requires the selection of a random number x from the Fp set to use as a private key. This value must fall between the range of 1 to n 1. First, the plaintext is changed into bits, and then those bits are mapped onto an elliptic curve in the form of (x, y) coordinates. After that, these points are transformed back into bits after going through the process of encryption. The following procedures are required in order to successfully encrypt data using an elliptic curve:

1. Initialization The encoding method is initialized by selecting appropriate parameters and initializing relevant data structures, such as E, Q, and p. This begins the process of decoding the input. This stage may, in some circumstances, call for the generation of random numbers or other types of beginning values.274
2. The Generation of Public Keys: It is necessary to generate a cryptographic key pair, which consists of a public key and a private key. Encrypting information requires the use of the public key, while decrypting it requires the use of the private key. It is computationally impossible to deduce the private key from the public key since the keys are generated in such a way that prevents this from happening. The private key, denoted by x, is utilized in the process of determining the public key, denoted by H, which is found by solving the equation H x.Q.
3. Encoding: After the generation of the public key, the plaintext data is encrypted with the public key. In the process of encoding, the original data are converted into a format that is unintelligible without the accompanying private key, as shown in Equation (4), where r is a random number. Encoding is a form of data encryption..

Ciphertext = Enc (data) = Ciphertext1 = r.Q

Ciphertext2 = data + r.H (4)

4. Decoding: The data that was encrypted are decrypted by making use of the private key, as demonstrated in Equation (5). The process of decoding involves reversing the steps of the encoding procedure so that the encrypted data can be converted back into its original format. The data can only be decrypted by the person who owns the private key, which ensures both its confidentiality and its security.

$\text{Dec (Ciphertext)} = \text{Ciphertext}_2 - x.\text{Ciphertext}_1 = \text{data} + r.H - x.r.Q = \text{data} + x.r.Q - x.r.Q = \text{data}$ (5)

4. The Proposed Model

It is important to generate a random private key in order to ensure the secure encoding and decoding of data using ECC. This can be accomplished by utilizing a random number generator. An strategy that makes use of ACS optimization to get such a key has been suggested as a means of satisfying this criteria. The degree of randomness in the key that is utilized during the encoding process has a considerable bearing on the quality of the encoding that is produced for sensitive data in ECC. The ACS is a mathematical method that looks for solutions in the search space, which can be thought of as being equivalent to the discrete nests or eggs that host birds lay. In this scenario, the purpose of the CS method is to locate the best possible private key (Pr) for ECC. This is so that it is possible to decipher the ciphertext back into its original plaintext in an accurate and risk-free manner. The viability of a solution, in this case a private key, can be ascertained by contrasting the decrypted plaintext with the original plaintext and calculating a viability value according to the precision and safety of the decoding. The following is one possible formulation for the objective function of the CS algorithm when applied in this setting:4. Decoding: The data that was encrypted are decrypted by making use of the private key, as demonstrated in Equation (5). The process of decoding involves reversing the steps of the encoding procedure so that the encrypted data can be converted back into its original format. The data can only be decrypted by the person who owns the private key, which ensures both its confidentiality and its security.

$f(Pr) = -h(Dec(CP, Pr), B)$ (6)

where CP is the ciphertext received after encoding, B is the original plaintext, Dec(CP, Pr) is the decoding of the ciphertext CP using the private key Pr, and h(x, y) is a function that measures the similarity or distance between the decrypted plaintext x and the original plaintext y. y is the plaintext that was originally encoded.

In this particular instance, the minus sign is utilized to change the fitness function into a minimization issue. The objective of this problem is to reduce as much as possible the gap that exists between the decrypted plaintext and the initial plaintext.

The model that is being presented includes each of the following steps:

1. The Aggressive Cuckoo Search technique begins with the setting of specific values, including the maximum number of iterations, the number of nests, the size of each nest, the chance of discovering foreign eggs, the step size, which is less than 0.01, and the chance of finding foreign eggs.

2. The eggs of the host bird are created by selecting a random sequence of numbers that is the same length as the

private key. The length of the private key used for ECC is 256 bits.

3. The equation (1) is used to produce new eggs, with the exception of the best egg, which does not change. If the new eggs are of higher quality than the existing ones, then one of them will be chosen at random using equation (6).

1. Dangerous nests are eliminated when alien eggs are discovered by a procedure that involves the generation of a random number and the comparison of that number to the probability of finding alien eggs (pa). In the event that the random number is lower than pa, a brand new solution is generated. If it is superior than the existing nests, it will replace them by moving somewhat further from where they are already positioned if it is successful.

2. The process of constructing new nests and locating foreign eggs is continued until either the maximum number of possible attempts is achieved or the quality of the nest that has shown to be the most successful reaches the desired standard. Experiments are performed in order to ascertain the values of n, pa.

The new technique is more efficient since it automatically assigns strings without requiring input from the user. It makes use of the CS algorithm in order to locate the private keys for users 1 and 2, and it derives the public key from Q in addition to the private key. The algorithm can do up to 175 iterations before reaching its limit.

The first algorithm sets up the variables and functions that will be used. To get started, the equation for the ECC curve is defined, and then point Q is calculated. The function CS() is responsible for locating the most suitable discrete nests or eggs for the private key during the encoding process. SHA-256 is then utilized to hash and encrypt the messages. During the decoding process, the hash function is applied initially. To modify the initial hash values, the XOR, and, right-shift, and left-shift methods are used. After that, ECC decoding is carried out.

The suggested method provides a number of advantages, including the automation of the string assignment process, a reduction in the amount of time required, and the guarantee of secure communication. In addition to this, it utilizes CS to optimize the solution and hash functions to add an additional layer of protection. In general, this new model is a method that is both effective and safe for the assignment of strings and the transmission of data between users.

The solution that has been offered uses ECC encoding with the XOR and shift-left operators in order to boost the data's level of confidentiality. Encoding (1), Encoding (2), Decoding (1), and Decoding (2) are the four phases that are based on ECC that are included in these operators. This not only makes the core of the model more robust but also guarantees that message delivery is safe. In order to decode the encrypted text, shift-left operations and XOR are performed on it first, and then the ECC method is applied.

Algorithm 1:

ECC():

a) Carry out the phase designated for initiation.

b) Determine a huge prime number using E and Fp as parameters, and report the result as p.

c) Choose which public parameters to use.

d) Construct a starting point designated as Q.

e) Utilize the CS() function to produce a private key, denoted by Pr.

f) Determine the value of the public key (Pu) by multiplying the value of the private key (Pr) by the base point (Q).

Encoding():

a) Extract the plaintext from the text file that is designated as B.

b) Utilizing the CS() function, produce a private key designated as PrB.

c) Convert the plaintext to an integer value, which will be referred to as m.

d) Produce a random number, denoted by r.

e) Compute the first portion of the ciphertext by multiplying the starting point, Q, by the random number, r.

f) Determine the second portion of the ciphertext2 by combining the plaintext (m) and the product of the private key (PrA) and the public key (Pu). This is accomplished by putting the two parts together.

g) Save the ciphertext as a pair consisting of the ciphertext1 and the ciphertext2 (CP) values.

h) To extract the encrypted text, it is necessary to calculate the SHA-256 hash of the plaintext.

Decoding():

a) To obtain the plaintext of the original encrypted text, calculate the SHA-256 hash of the encrypted text.

b) Acquire the ciphertext, also known as the CP.

c) Extract the portion of the ciphertext1 that is on the left.

d) Take out the appropriate section of the ciphertext2.

e) Acquire the initial plaintext (m) by deducting from the second half of the ciphertext (ciphertext2) the product of the private key (PrA) and the starting point (ciphertext1).

f) Convert the integer value of the plaintext, which is m, to the plain text value that corresponds to it.

```
CS():
a) Begin by initializing the existing solutions.
Algorithm 1 (continued):
b) Begin the process of a loop.
c) Determine the level of fitness possessed by each individual solution within the population of size n.
d) Determine which solution is the most effective.
e) For each solution in the population, take one step closer to the optimal solution while calculating the size  of the step using a random number that follows a normally distributed distribution.
f) In order to obtain the new answer, round the result of the step off to the next whole number.
g) The answer should not deviate too much from the parameters of the problem.
h) Determine whether or not the proposed alternative is viable.
i) If the applicability of the new solution is superior to the applicability of the prior solution, the new solution should be accepted.
j) From the population of possible solutions, pick two at random.
k) Produce a scaling factor that is chosen at random.
l) Generate a new candidate solution by adding the scaled difference between the two solutions to each solution in the population. This will result in the generation of a new candidate solution.
m) Determine whether or not the candidate solution is suitable.
n) You should go ahead and accept the candidate solution if the fitness of the new solution is greater than the fitness of the old solution.
o) Perform a random permutation on the population.
p) Carry out one more random permutation on the population.
q) Produce a number chosen at random.
r) Determine whether the potential solution meets the criteria.
s) Produce yet another number chosen at random.
t) The candidate solution should be accepted if the random number is lower than a predetermined threshold of probability (pa).
u) The answer must remain constrained by the parameters of the problem.
v) Determine whether or not the best solution is practical.
w) Keep going around in the loop until either the maximum number of possible iterations is reached or the required level of physical fitness is obtained.
x) Determine whether or not the best solution is feasible.
y) Exit the feedback loop.
```

```
SHA-256():
a) You will need to initialize the hash value.
b) Begin by setting the values of the round constants.
c) You need to initialize the XOR operator as well as the shift right operator.
d) Set the values of the working variables to their defaults
a) Begin processing data using the compression function's main loop.
b) Include the chunk that has been compressed in the current hash value.
c) Generate the final value for the hash.
d) Bring the function to a close.
```

5. Evaluation and Results

The purpose of this part is to assess the throughput and speed of the proposed model in terms of both encoding and decoding. The experiments were carried out on a computer running Windows 10 Professional, with the C# programming language and Microsoft Visual Studio 2012 being the tools of choice. The personal computer included a 1.60 GHz AMD E-350 processor and 4 gigabytes of random-access memory (RAM). The length of the key played a factor in determining the initial population size, which was determined in a random manner by the algorithm. For this particular iteration of the model, the key length was figured out by picking random integers from the range of 0 to 127. Because of this, the size of the starting population was determined by the length of this key. These experimental conditions served as the foundation for the analysis of the encoding and decoding procedure's overall performance.

5.1. Performance Analysis

5.1.1. Decoding and Encoding Times

In this section, we will evaluate the suggested model's encoding and decoding speeds so that you can make an informed decision. The results are presented in Figs. 1 and 2, which are numbered accordingly. The encoding time (in milliseconds) of the proposed model at various iterations with an initial population of 20 is shown graphically in Fig. 1. According to the findings, the amount of time needed for encoding decreases as the number of iterations grows, and it reaches its smallest value after 75 iterations.

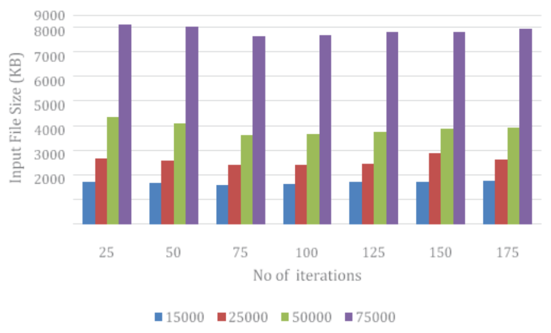

Fig 1: Iteration-based encoding of the passage of time

Encoding a file that is 75,000 KB in size, for instance, will take 8,125 milliseconds for the first 25 repetitions, 7,635 milliseconds for the next 75 iterations, and 7,930 milliseconds for the next 175 iterations. The findings are backed up by the data that is shown in the figures, which were acquired through an exhaustive experimental research that was carried out on the proposed model.

On the contrary, Fig. 2 displays the decoding time of the suggested model in milliseconds through a variety of iterations while using a population of 20 as the starting point for the data. When the amount of time required for decoding is less than the amount of time required for encoding, the system will function in the most timely and effective manner possible. According to the data, the amount of time required for decoding reaches its lowest point after 125 iterations. For example, the time required to decrypt a file that is 75,000 KB takes 1123.201 milliseconds for the first 25 rounds, 1102.012 milliseconds for the next 125 iterations, and 1237.072 milliseconds for the final 175 iterations.

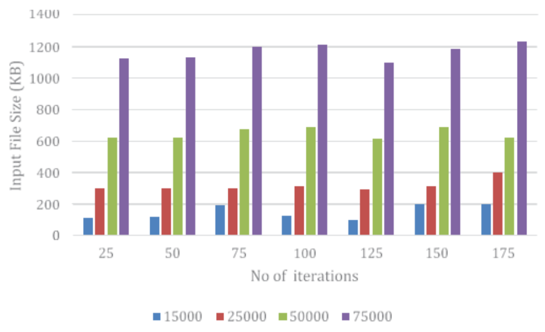

Fig 2: Time required for decoding based on iterations

According to the results depicted in Fig. 3 Mean encoding and decoding times vary across iterations in the proposed approach. At the 75th iteration, the encoding time is the most efficient, while at the 125th iteration, the decoding time is the most efficient. These results show that the suggested model provides the required level of security while also improving execution speed. Therefore, it follows that the proposed paradigm is a practical means to safe and effective data encoding and decoding.

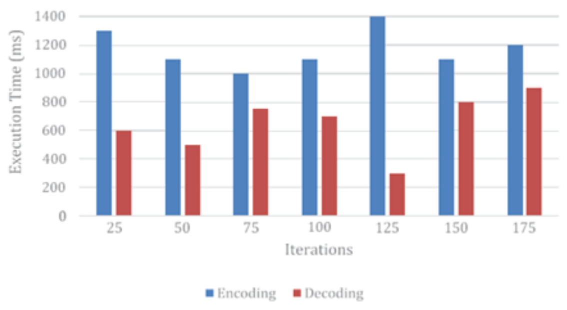

Fig 3: Encoding and decoding times on average for different iterations

This evaluation shows that the proposed model works well in terms of encoding and decoding speed. Minimum encoding and decoding times can be achieved with 75 and 125 iterations, respectively. These findings will aid in the creation of more secure and reliable encoding techniques, which will further cryptography's progress. In addition, the proposed model's faster execution time can be useful for a wide range of applications, particularly those dealing with huge files.

5.2. Comparison and Evaluation

5.2.1. Encoding and Decoding Based on AES-RSA and Proposed Model

Using Visual Studio, a comparative study of the proposed model and AES-RSA [41] is shown in Table 1. This analysis is based on the examination of 14 files with varied sizes and a key length of 128 bits. The primary purpose of this investigation was to evaluate how efficient the proposed model is in comparison to AES-RSA in terms of the amount of time required for encoding and decoding. According to the findings, the suggested model offers superior performance in comparison to AES-RSA, in particular while working with a data size of 50 MB. The time it takes to encode data using AES-RSA in this case is 6.53 seconds, while the time it takes to encode data using the suggested model is just 3.29 seconds. This demonstrates a considerable improvement in processing time.

Table 1: An evaluation of the suggested model in contrast to AES-RSA

File Size (MB)	AES-RSA		Proposed Model	
	Encoding	Decoding	Encoding	Decoding
1	1.73	18.92	0.89	0.77
3	2.01	19.72	1.12	1.03
7	2.05	18.49	1.36	1.27
10	3.01	21.08	1.94	1.90
17	3.61	21.91	2.15	2.03

21	3.72	29.01	2.37	2.45
25	3.73	32.28	2.40	2.13
28	4.57	32.01	2.60	2.38
32	4.42	31.08	2.45	2.65
35	5.29	28.1	3.45	2.67
39	4.97	27.93	3.45	2.76
42	5.52	27.88	3.45	3.65
46	5.62	29.73	3.46	3.89
50	6.53	31.59	4.32	3.09

The RSA cryptography system is well-known for being a heavyweight due to the huge key size it uses. In contrast, the ECC provides an alternative that is more efficient and uses a smaller key size. As a consequence, processing times are reduced while less memory is utilized [42]. As a result, the ECC can be implemented on devices that have a restricted amount of resources, which results in faster computations in real time.

According to the findings that are depicted in Figure 4, a comparison was made between the amount of time required to encode using the suggested model and that required by AES-RSA. According to the findings, the encoding time of the suggested model is noticeably shorter, which results in a decrease of approximately 29.24% in the amount of time required for the execution of encoding on a device of File Size (50 MB). Furthermore, It has been demonstrated that the suggested model outperforms AES-RSA, showing its superiority over the present technique that uses AES-RSA in terms of efficiency.

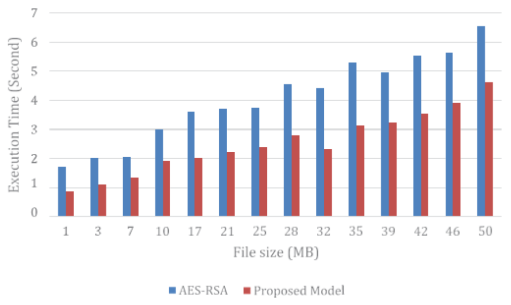

Fig 4: A comparison of the suggested model’s and AES-RSA’s encoding times

Figure 5 presents a comparison of the amount of time required to decode the proposed model with that of the AES-RSA algorithm. This comparison demonstrates that the suggested model surpasses AES-RSA in terms of speed. The slower performance of the RSA technique can be attributed to the fact that its encoding and decoding processes are extremely computationally costly. These processes entail modular exponentiation operations and computations involving the private key exponent. Because of the high amount of computations that are required for these procedures, the processing speed will be decreased. In contrast to RSA, the suggested model involves fewer computations during both the encoding and the decoding processes; as a result, the proposed model is capable of significantly increased processing rates. The decreased amount of computing work needed to run the proposed model is consequently to blame for the model's higher performance.

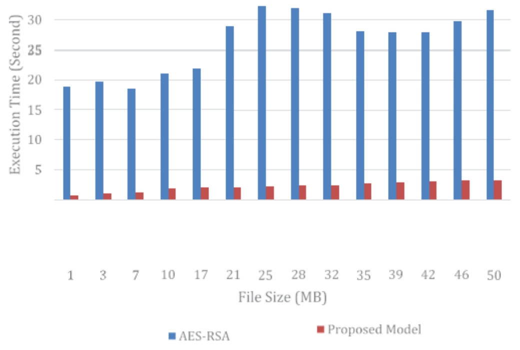

Fig 5: Comparing the decoding times of the recommended model versus AES-RSA is something that has to be evaluated.

In a nutshell, the model proposed in this research outperforms AES-RSA in terms of the speed at which it encodes and decodes data. As a result, it demonstrates itself to be a more effective and workable alternative for use in cryptographic applications, particularly in circumstances in which there are restricted resources. The application of ECC within the proposed model reduces the weight of calculation, so making it faster and better suitable for situations that take place in real time. Due to these benefits, the model that was developed is an attractive candidate for a solution to the problem of secure data exchange.

5.2.2. FA, GA, and Proposed Model-Based Encoding and Decoding

This section gives a comparative examination of two metaheuristic algorithms, specifically the Firefly Algorithm (FA) [43] and the Genetic Algorithm (GA) [44], in connection with the suggested model for the generation of ECC keys. The search for the best ECC solution is carried out using these algorithms, which make use of selection, crossover, and mutation operators. According to the findings, which are summarized in Table 2, the proposed CS algorithm performs better than both FA and GA in terms of the amount of time required for encoding and decoding, which ultimately results in improved solutions, search, and power convergence. To be more explicit, the findings indicate that the suggested CS algorithm demonstrates shorter encoding and decoding times than either FA or GA, despite the fact that FA performs better than GA. However, FA performs better than GA. For example, encrypting a file that is 10,000 KB with

ECC-FA-SHA-256 and ECC-GA-SHA-256 takes 4224 and 4370 milliseconds, respectively, while decoding the same file with those algorithms takes 4151 and 4292 milliseconds, respectively. In comparison, the encoding and decoding timings for the model that was proposed came in at 4011 and 3726 milliseconds, respectively.

The findings that are provided in Figures 6 and 7 offer additional confirmation that the proposed model is superior to alternative methods in terms of both the amount of time it takes to execute the algorithm and the amount of efficiency it achieves. When placed in the context of the CS algorithm, the existing strategies for generating ECC keys may be contrasted with the model that was suggested, and the conclusion that can be drawn is that the latter is the superior option. These findings provide solid evidence that putting into practice the CS algorithm that was suggested results in an increase in both the security and efficiency of cryptographic systems.

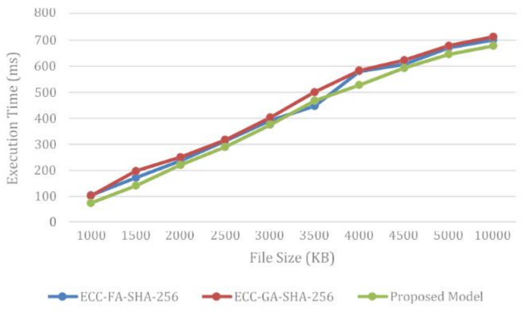

Fig 6: A comparison of the encoding time of the suggested model with that of GA and FA is shown here.

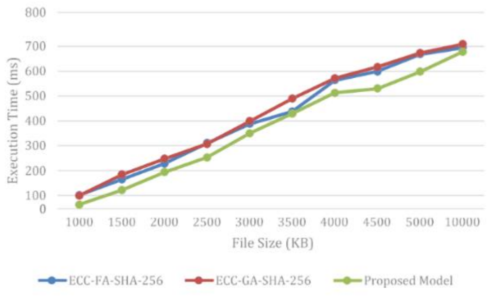

Fig 7: Demonstrates a contrast between the amount of time required to decode the suggested model and the times required by GA and FA

5.3. Throughput

In this investigation, the suggested model was evaluated in terms of its capacity for encoding and decoding, with speed serving as the criterion for evaluation. Speed was calculated by applying equations (7) and (8) [45]. A better indicator of a superior model was one with a higher measurement. For the purpose of determining how well the model functions, it was applied to a file that was 55000 KB in size, and the results were compared to those obtained by the (ECC-FA-SHA-256) and (ECC-GA-SHA-256) models. Encoding and decoding throughputs of 13.71 and 14.76 milliseconds, respectively, were demonstrated by the suggested paradigm. The (ECC-FA-SHA-256) model, on the other hand, had encoding and decoding throughputs of 13.02 and 13.24 ms, while the (ECC-GA-SHA-256) model had throughputs of 12.58 and 12.81 ms, respectively, for both processes. Therefore, in terms of both encoding and decoding speed, the model that was provided was superior to both of the models that were used for comparison.

Encoding Throughput = Σ(Input File)/Σ(Encoding Time) (7)

The throughput of decoding = Σ(Input File)/Σ(Decoding Time) (8)

The encoding and decoding throughputs of the suggested model were discovered to be approximately 5.31% and 11.41% better than those of the model (ECC-FA-SHA-256), respectively. In a similar vein, the encoding and decoding throughputs of the suggested model were roughly 8.95% and 15.17% better than those of the model (ECC-GA-SHA-256), respectively. According to these findings, the suggested model is substantially more efficient and effective than the other models when it comes to encoding and decoding data in a timely manner while maintaining a high level of security.

Table 2: Presents a comparison of the encoding and decoding times (in milliseconds) of the proposedmodel with those of the (FA) and (GA)

File Size (KB)	ECC-FA-SHA-256		ECC-GA-SHA-256		Proposed Model	
	Encoding	Decoding	Encoding	Decoding	Encoding	Decoding
1000	105	101	103	99	78	64
2000	172	164	198	183	147	129
3000	237	228	251	247	230	198
4000	313	310	317	307	296	258

5000	391	387	402	398	377	355
6000	447	438	501	489	468	435
7000	580	563	583	571	534	518
8000	607	599	623	617	597	533
9000	671	667	679	673	645	599
10000	701	694	713	708	679	688
55000	4224	4151	4370	4292	4021	3735
Throughput	13.02083333	13.24981932	12.58581236	12.81453868	13.7122986	14.76113854

6. Conclusion & Future Works

The Internet of Things (IoT) has quickly become ingrained in many different facets of modern society's way of life. Despite this, securing the security of IoT devices continues to be a big concern due to the limited resources available on these devices. It is essential to protect the privacy and security of the data that is transmitted by IoT devices, which can include text messages, photos, and sounds. It is also essential to ensure that this data is not altered in any way. In order to achieve this objective, particular cryptographic algorithms perform more effectively than others do in terms of their efficiency. In this work, a novel technique to securing data that is conveyed by small Internet of Things-based door locks is presented. The approach makes use of ECC and SHA-256 in conjunction with the ACS algorithm. The method that has been suggested is effective, and it guarantees that encoding and decoding data will take significantly less time. According to the findings of the inquiry, the most effective method for encoding requires 75 iterations, and the most effective method for decoding requires 125 iterations. The model that was developed is 15,17% more efficient than other methods, and it offers acceptable security against attacks; as a result, it is suitable for devices that have a limited amount of resources to work with. Simulations were run as part of this work to test the viability of the suggested method, and it is anticipated that further study will investigate further approaches to the generation of private keys.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] L. Kong and B. Ma, "Intelligent manufacturing model of construction industry based on Internet of Things technology," The International Journal of Advanced Manufacturing Technology, vol. 107, pp. 1025–1037, 2020.

[2] Jha, A. Athanerey and A. Kumar, "Role and challenges of internet of things and informatics in healthcare

[3] research," Health and Technology, vol. 12, no. 4, pp. 701–712, 2022.

[4] Kamilaris and A. Pitsillides, "Mobile phone computing and the internet of things: A survey," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 885–898, 2016.

[5] P. M. Chanal and M. S. Kakkasageri, "Security and privacy in IoT: A survey," Wireless Personal Communications, vol. 115, pp. 1667–1693, 2020.

[6] K. R. Sarkar, "Assessing insider threats to information security using technical, behavioural and organisa-tional measures," Information Security Technical Report, vol. 15, no. 3, pp. 112–133, 2010.

[7] Makhdoom, M. Abolhasan, H. Abbas and W. Ni, "Blockchain's adoption in IoT: The challenges, and a way forward," Journal of Network and Computer Applications, vol. 125, pp. 251–279, 2019.

[8] Ghani, K. Mansoor, S. Mehmood, S. A. Chaudhry, A. U. Rahman et al., "Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key," International Journal of Communication Systems, vol. 32, no. 16, pp. e4139, 2019.

[9] R. Aldawira, H. W. Putra, N. Hanafiah, S. Surjarwo and A. Wibisurya, "Door security system for home monitoring based on ESP32," Procedia Computer Science, vol. 157, pp. 673–682, 2019.

[10] Ha, "Security and usability improvement on a digital door lock system based on internet of things,"

[11] International Journal of Security and its Applications, vol. 9, no. 8, pp. 45–54, 2015.

[12] Jacobsson, M. Boldt and B. Carlsson, "A risk analysis of a smart home automation system," Future Generation Computer Systems, vol. 56, pp. 719–733, 2016.

[13] M. Ahtsham, H. Y. Yan and U. Ali, "IoT based door lock surveillance system using cryptographic

algorithms," in 2019 IEEE 16th Int. Conf. Networking, Sensing and Control (ICNSC), Banff, AB, Canada, pp. 448–453, 2019.

[14] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, pp. 210–214, 2009.

[15] V. S. Miller, "Use of elliptic curves in cryptography," in 12th Int. Workshop, Canada, SAC, pp. 417–426, 2005.

[16] H. Yoshida and A. Biryukov, "Analysis of a SHA-256 variant," in 12th Int. Workshop, Canada, SAC, pp. 245–260, 2006.

[17] S. R. Prasanna and B. S. Premananda, "Performance analysis of MD5 and SHA-256 algorithms to maintain data integrity," in 2021 Int. Conf. on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, pp. 246–250, 2021.

[18] N. K. Sreelaja and G. V. Pai, "Stream cipher for binary image encryption using ant colony optimization based key generation," Applied Soft Computing, vol. 12, no. 9, pp. 2879–2895, 2012.

[19] Shankar and P. Eswaran, "A secure visual secret share (VSS) creation scheme in visual cryptography using elliptic curve cryptography with optimization technique," Australian Journal of Basic & Applied Science, vol. 9, no. 36, pp. 150–163, 2015.

[20] Shankar and P. Eswaran, "An efficient image encryption technique based on optimized key generation in ECC using genetic algorithm," in Artificial Intelligence and Evolutionary Computations in Engineering Systems: Proc. of ICAIECES 2015, India, Springer, pp. 705–714, 2016.

[21] Dworak, J. Nalepa, U. Boryczka and M. Kawulok, "Cryptanalysis of SDES using genetic and memetic algorithms," in Recent Developments in Intelligent Information and Database Systems, Springer, Cham, pp. 3–14, 2016.

[22] Kalaiselvi and A. Kumar, "An empirical study on effect of variations in the population size and generations of genetic algorithms in cryptography," in 2017 IEEE Int. Conf. on Current Trends in Advanced Computing (ICCTAC), Bangalore, India, pp. 1–5, 2017.

[23] Ahmad, M. Z. Alam, Z. Umayya, S. Khan and F. Ahmad, "An image encryption approach using particle swarm optimization and chaotic map," International Journal of Information Technology, vol. 10, pp. 247–255, 2018.

[24] S. Kota, V. N. R. Padmanabhuni and K. Budda, "Authentication and encryption using modified elliptic curve cryptography with particle swarm optimization and cuckoo search algorithm," Journal of the Institution of Engineers (India): Series B, vol. 99, no. 4, pp. 343–351, 2018.

[25] R. Kiruba and T. Sree Sharmila, "Secure data hiding by fruit fly optimization improved hybridized seeker algorithm," Multidimensional Systems and Signal Processing, vol. 33, no. 3, pp. 1423–1443, 2020.

[26] T. Avudaiappan, R. Balasubramanian, S. S. Pandiyan, M. Saravanan, S. K. Lakshmanaprabu et al., "Medical image security using dual encryption with oppositional based optimization algorithm," Journal of Medical Systems, vol. 42, pp. 1–11, 2018.

[27] S. Mitra, G. Mahapatra, V. E. Balas and R. Chattaraj, "Public key cryptography using harmony search algorithm," in Innovations in Infrastructure, Singapore, Springer, pp. 1–11, 2019.

[28] Shankar, M. Elhoseny, E. Perumal, M. Ilayaraja and K. Sathesh Kumar, "An efficient image encryption scheme based on signcryption technique with adaptive elephant herding optimization," in Cybersecurity and Secure Information Systems: Challenges and Solutions in Smart Environments. Springer, Cham, pp. 31–42, 2019.

[29] Din, S. K. Pal and S. K. Muttoo, "Applying PSO based technique for analysis of geffe generator cryptosystem," in Harmony Search and Nature Inspired Optimization Algorithms: Theory and Applications, ICHSA 2018. Singapore: Springer, pp. 741–749, 2019.

[30] S. Vimal, M. Khari, R. G. Crespo, L. Kalaivani, N. Dey et al., "Energy enhancement using multiobjective ant colony optimization with double Q learning algorithm for IoT based cognitive radio networks," Computer Communications, vol. 154, pp. 481–490, 2020.

[31] Z. Wang, D. Liu and A. Jolfaei, "Resource allocation solution for sensor networks using improved chaotic firefly algorithm in IoT environment," Computer Communications, vol. 156, pp. 91–100, 2020.

[32] P. Siddavaatam and R. Sedaghat, "A novel multi-objective optimizer framework for TDMA-based medium access control in IoT," CSI Transactions on ICT, vol. 8, pp. 319–330, 2020.

[33] K. Tamilarasi and A. Jawahar, "Medical data security for healthcare applications using hybrid lightweight encryption and swarm optimization algorithm," Wireless Personal Communications, vol. 114, pp. 1865– 1886, 2020.

[34] S. Yousefi, F. Derakhshan, H. S. Aghdasi and H. Karimipour, “An energy-efficient artificial bee colony- based clustering in the internet of things,” Computers & Electrical Engineering, vol. 86, pp. 106733, 2020.

[35] P. Verma, N. Jain and S. K. Pal, “Design and analysis of an optimal ECC algorithm with effective access control mechanism for big data,” Multimedia Tools and Applications, vol. 79, pp. 9757–9783, 2020.

[36] S. Nandan, S. Singh and L. K. Awasthi, “An efficient cluster head election based on optimized genetic algorithm for movable sinks in IoT enabled HWSNs,” Applied Soft Computing, vol. 107, pp. 107318, 2021.

[37] Mullai and K. Mani, “Enhancing the security in RSA and elliptic curve cryptography based on addition chain using simplified swarm optimization and particle swarm optimization for mobile devices,” International Journal of Information Technology, vol. 13, pp. 551–564, 2021.

[38] S. K. Mousavi and A. Ghaffari, “Data cryptography in the internet of things using the artificial bee colony algorithm in a smart irrigation system,” Journal of Information Security and Applications, vol. 61, pp. 102945, 2021.

[39] D. W. Sims, D. Righton and J. W. Pitchford, “Minimizing errors in identifying Lévy flight behaviour of organisms,” Journal of Animal Ecology, vol. 76, no. 2, pp. 222–229, 2007.

[40] Lévy, “Théorie de l’addition des variables aléatoires,” Bulletin de la Société Mathématique de France, vol. 67, pp. 1–41, 1939.

[41] X. S. Yang and S. Deb, “Engineering optimisation by cuckoo search,” International Journal of Mathematical Modelling and Numerical Optimisation, vol. 1, no. 4, pp. 330–343, 2010.

[42] Conti, N. Dragoni and V. Lesyk, “A survey of man in the middle attacks,” IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2027–2051, 2016.

[43] L. Zou, M. Ni, Y. Huang, W. Shi and X. Li, “Hybrid encryption algorithm based on AES and RSA in file encryption,” in Frontier Computing: Theory, Technologies and Applications (FC 2019), vol. 8. Singapore: Springer, pp. 541–551, 2020.

[44] Bafandehkar, S. M. Yasin, R. Mahmod and Z. M. Hanapi, “Comparison of ECC and RSA algorithm in resource constrained devices,” in 2013 Int. Conf. on IT Convergence and Security (ICITCS), Macao, China, pp. 1–3, 2013.

[45] X. S. Yang, “Firefly algorithms for multimodal optimization,” in Int. Symp. on Stochastic Algorithms, Sapporo, Japan, pp. 169–178, 2009.

[46] J. H. Holland, “Genetic algorithms,” Scientific American, vol. 267, no. 1, pp. 66–73, 1992.

[47] Lemma, M. Tolentino and G. Mehari, “Performance analysis on the implementation of data encryption algorithms used in network security,” International Journal of Computer and Information Technology, vol. 4, no. 4, pp. 711–717, 2015.

[48] Mwitia, Shawn Muthomi, and Davies Rene Segera. "An Aggressive Cuckoo Search Algorithm for Optimum Power Allocation in a CDMA-Based Cellular Network." The Scientific World Journal 2022 (2022).

[49] Sabonchi, Arkan Kh Shakr, and Zainab Hasim Obaid. "Ensuring Information Security in Smart Door Lock Systems Using the Cuckoo Search Algorithm."

[50] Srinivasarao, Popuri, and Aravapalli Rama Satish. "A Novel Hybrid Optimization Algorithm for Materialized View Selection from Data Warehouse Environments." Computer Systems Science & Engineering 47.2 (2023).