Journal of
*Sensor and
Actuator Networks*

Article

Hybrid Encryption Model for Secured Three-Phase Authentication Protocol in IoT

Amr Munshi * and Bandar Alshawi

Department of Computer and Network Engineering, College of Computing, Umm Al-Qura University, Makkah 24382, Saudi Arabia; bmhshawi@uqu.edu.sa
* Correspondence: aaamunshi@uqu.edu.sa

Abstract: The Internet of things (IoT) has recently received a great deal of attention, and there has been a large increase in the number of IoT devices owing to its significance in current communication networks. In addition, the validation of devices is an important concern and a major safety demand in IoT systems, as any faults in the authentication or identification procedure will lead to threatening attacks that cause the system to close. In this study, a new, three-phase authentication protocol in IoT is implemented. The initial phase concerns the user registration phase, in which encryption takes place with a hybrid Elliptic Curve Cryptography (ECC)–Advanced Encryption Standard (AES) model with an optimization strategy, whereby key generation is optimally accomplished via a Self-Improved Aquila Optimizer (SI-AO). The second and third phases include the login process and the authentication phase, in which information flow control-based authentication is conducted. Finally, decryption is achieved based on the hybrid ECC–AES model. The employed scheme's improvement is established using various metrics.

Keywords: IoT; authentication; registration phase; information flow; encryption

check for updates

Citation: Munshi, A.; Alshawi, B. Hybrid Encryption Model for Secured Three-Phase Authentication Protocol in IoT. *J. Sens. Actuator Netw.* **2024**, *13*, 41. https://doi.org/10.3390/jsan13040041

Academic Editor: Lei Shu

Received: 4 April 2024
Revised: 23 June 2024
Accepted: 6 July 2024
Published: 17 July 2024

1. Introduction

The Internet of things (IoT) is a dominant communication theory associated with various areas of use, such as monitoring, e-health, and smart grid appliances [1–3]. IoT is relevant to many aspects of everyday life, such as the functioning of smart cities, the military, smart grid development, traffic, healthcare, etc. The IoT system encompasses many interrelated IoT sensors or smart devices that converse over the internet. These independent devices are employed in a variety of complicated fields to collect and sense data or to carry out selected activities [4,5]. Usually, in IoT cloud-oriented schemes, the gathered data from IoT sensors is passed to cloud servers to be stored. Afterwards, certified users can access the required data from the related cloud server database. Nevertheless, in numerous other circumstances, the users might need direct, instantaneous data from the sensor [6–8].

The number of IoT-oriented schemes is rising quickly, owing to the huge variety of IoT devices and their practical manufacturing costs. In addition, cloud technology thrives on the computational power capabilities of IoT applications [9–11]. Therefore, cloud-oriented schemes in the IoT, with realistic access, help governments, organizations, and specialists to capably handle their sources, offer timely and precise data, and decrease human participation. Nevertheless, security concerns are quickly rising alongside the extensive implementation of IoT-oriented services. Therefore, there is an essential requirement to model mature safety solutions for protecting contemporary IoT schemes from potential risks [12–14].

The development of electronic commerce and data security depends on our ability to safeguard information. Probably the most crucial technology for data protection is cryptography. There are two main types of encryption, namely symmetric and asymmetric encryption. Asymmetric encryption employs two keys—a public key and a private key—to encrypt and decrypt data, as opposed to symmetric encryption, which uses a single secret key. Due to its high efficiency in encrypting large texts, symmetric key encryption is adopted in this study. The Advanced Encryption Technique (AES) is a symmetric key encryption standard that is commonly used to secure data where data secrecy is a crucial and pressing concern. Key management for ECC, which is appropriate for key encryption and digital signature, is simple. This work proposes a mixed encryption architecture based on ECC and AES that uses ECC to encrypt and transfer the AES key and, as a result, AES to encrypt communication data.

A wide variety of research has been proposed for mutual validation in diverse circumstances for IoT design, which makes use of fundamental cryptographic equipment such as asymmetric/symmetric encryption, hash operations, and Elliptic Curve Cryptography (ECC) [15,16]. The ECC model was developed by Neal and Victor in 1985 for cryptography purposes [17]. Arij et al. developed a solution that relied on the fog servers and fog users to validate one another secretly. However, the developed model did not consider a secured method to ensure the user's intractability [18,19].

The contributions to the current work are as follows:

1. Present a novel secured authentication model for IoT.
2. Adaptation of an optimized hybrid Elliptic Curve Cryptography (ECC)—Advanced Encryption Standard (AES) model for encryption.
3. Propose a novel Self-Improved Aquila Optimizer (SI-AO) model for selecting the optimal private keys.

The paper is arranged as follows: Section 2 reviews the previous relevant work. Section 3 briefly explains the processes involved in the secured authentication scheme. Section 4 explains SI-AO-based optimization for optimal key selection and lists all the steps in the proposed model. Further, Section 5 illustrates outcomes, and the conclusions are given in Section 6.

2. Literature Review

In 2020, Minahil et al. [20] introduced an enhanced and more secure distant user-authenticated protocol to address various security weaknesses. Furthermore, the developed model was safer not only against user imitation attacks but also against safety attacks. It had realistic communication, storage, and computation costs and was an enhanced candidate for employment in IoT networks. Melki et al. [21] proposed a light-weighted secured authentication model for IoT. The method depended upon two conceptions: "configurable Physical Unclonable Functions (PUF) within IoT devices, and channel-based parameters". The scheme provided higher robustness in opposition to dissimilar attack types while sustaining lower intricacy. Further, an enhanced novel light-weighted hash-chain-oriented and forward-secured authentication method in healthcare IoT was presented by Mahdi et al. [22]. Alzahrani et al. [23] modeled an enhanced "Lightweight Authentication Scheme for IoT Deployments (ILAS-IoT)" for securing the IoT from attacks. ILAS-IoT performed the procedure accurately by minimizing communication and computational overheads. The modeled approach also resisted all recognized and stolen verifier attacks that were obvious from informal and formal security studies. Subsequently, Khalid et al. [24] proposed a decentralized authentication and access control mechanism for lightweight IoT. Their approach demonstrated improved performance among existing techniques.

Building upon these advancements, in 2021, Ahmed et al. [8] adopted higher scalability models with a proficient user registration procedure, where the legal user accessed the recently added system entity with no further processes. Additionally, a fuzzy extractor model was deployed on the user side to verify the user's biometric data. Eventually, more required characteristics were provided, and mutual confirmation was achieved with lower communication and computational costs over existing models. Subsequently, in the same year, Das et al. [25] examined a smart-card-based, remote, secure, and lightweight authentication scheme and showed that their method was unsecured in opposition to severe attacks, including "privileged-insider attack, stolen smart card attacks, Ephemeral

Secret Leakage (ESL) attacks, password change attack and user impersonation attacks". To address these safety drawbacks, some solutions were offered to help build a more effective and secure user verification method to protect the next generation of the IoT. The common security problems of guaranteed anonymous mutual validation and trust registration were studied in [26].

A new authentication system suitable for IoT contexts that fixes these security issues was proposed by Son et al. [27]. They developed an approach that only utilizes hash and exclusive-or operations. In addition, Ehui et al. [28] presented a mutual authentication system for the IoT. The protocol adopted simple cryptography methods to establish safe mutual authentication between the sensor node and gateway. The Barrows–Abadi–Needham (BAN)-logic technique was used to examine the protocol, and the findings revealed that the suggested scheme attained good security and performance when compared to related current protocols. In this work, we demonstrate that the suggested protocol offers higher security and performance when contrasted with current authentication protocols by analyzing the protocol using informal and formal analysis methods, including the BAN logic, real-or-random (ROR) model, and the AVISPA simulation. Hence, we found that the suggested protocol is viable and appropriate for actual IoT environments. Table 1 shows the reviews of conventional authentication schemes in the IoT.

Table 1. Reviews of conventional authentication schemes in the IoT.

Author	Deployed Schemes	Features	Challenges
Minahil et al. [20]	CK-adversary model	Higher security Safeguards anonymity	Need to consider the practical implementation
Ahmed et al. [8]	Fuzzy scheme	Less overhead Minimal cost	Needs deliberation on blockchain technologies
Das et al. [25]	Secure and lightweight authentication scheme	Minimizes cost Enhanced security	Needs research on machine-to-machine (M2M) security schemes
Mahdi et al. [22]	Real-or-Random (ROR) scheme	Minimal cost Secure and efficient	Should focus on improving execution time.
Alzahrani et al. [23]	Fuzzy Probabilistic Generation	Minimal complexity Minimal overhead	Stolen verifier attack should be considered.
Khalid et al. [24]	Elliptic Curve Digital Signature Algorithm (ECDSA)	Needs minimal power Reduced power	Lightweight consensus score was not computed
Jebri et al. [26]	ECC	Ensures trust Minimal time cost	Needs consideration on computing resources
Melki et al. [21]	ROR model	Low cost Higher robustness	Requires more time
Son et al. [27]	Hash exclusive-or operations	Higher security Higher performance	Mostly suiTable in IoT contexts
Ehui et al. [28]	Simple mutual authentication system	Good security Good performance	Requires a suiTable technique to assess protocol

2.1. Problem Statement

Numerous methods have focused on authentication protocols in the IoT. However, common problems persist, such as time consumption, security issues, the need for consideration of standard encryption algorithms, machine-to-machine (M2M) security schemes, and computing resource constraints. To address these challenges, this paper proposes a secured authentication protocol in IoT using metaheuristic optimization algorithms. However, authentication remains a critical limiting factor for IoT deployment due to many reasons, including the fact that the implementation of robust authentication mechanisms requires additional hardware and software development. This leads to higher costs, not to mention the continuous maintenance that adds to that cost. IoT systems comprise devices requiring secure authentication, making it challenging to ensure that authentication protocols can scale to accommodate larger networks. Robust authentication protects sensitive data, ensures that only authorized devices can access the IoT network, establishes trust and reliability among IoT devices and users, and is crucial for compliance with security and privacy regulations. Furthermore, secure authentication protocols ensure that devices from different manufacturers can communicate seamlessly. Accordingly, authentication is considered a significant limiting factor for IoT deployment due to its impact on cost, power consumption, and scalability. Addressing these challenges is crucial for the secure and efficient operation of IoT systems. Therefore, approaches that aim to provide a secure, efficient, and scalable authentication protocol contribute to the advancement of IoT technologies.

2.2. Objectives

The main objectives of this research are as follows:

- One objective is to overcome the aforesaid challenges by proposing a hybrid encryption model for a secured, three-phase authentication protocol (registration phase, login phase, and authentication phase).
- To achieve this, we optimally generate the key using the metaheuristic method.

3. Processes Involved in Secured Authentication Schemes in IoT

The adopted protocol includes servers and users. The agreement encompasses three stages: "user registration, login, and authentication". The user corresponds to the major contributor to the communication, and the server corresponds to the entity that communicates with users.

User registration:

To register, the user must first create a new user account, which is a database record that describes how they will authenticate their identity.

Login:

The user must enter their login and password to access a computer.

Authentication process:

To authenticate, the user must provide proof that their identity matches that of their user account. The user registration, login, and authentication are together known as the authentication protocol.

3.1. State of the Art in Secured Authentication Schemes in the IoT

The developed authentication protocol includes three vital phases:

- Initially, the registration phase is carried out, where encryption is accomplished with a hybrid ECC–AES model.
- Subsequently, optimal key selection is performed via SI-AO to choose the best private keys in AES.
- Further, the login and authentication phases are performed, where information flow control-oriented authentication is conducted.
- Finally, decryption is accomplished using a hybrid ECC–AES model. Figure 1 shows the overall depiction of the suggested SI-AO-oriented model.

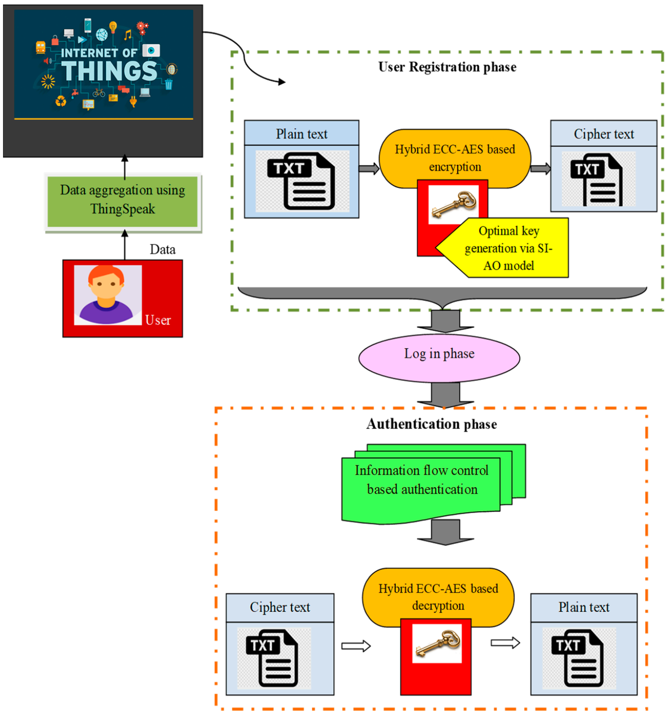

INTERNET OF THINGS
Data aggregation using ThingSpeak
Data
User
User Registration phase
Plain text
TXT
Hybrid ECC-AES based encryption
Cipher text
TXT
Optimal key generation via SI-AO model
Log in phase
Authentication phase
Information flow control based authentication
Cipher text
TXT
Hybrid ECC-AES based decryption
Plain text
TXT

Figure 1. Overall depiction of the suggested SI-AO-oriented model.

3.2. User Registration Phase

Initially, the user V_c chooses an ID id_c, password pw_c, bio info r_i, and an arbitrary number n, which is shown in Equation (1). Then, the registration process starts, as mentioned below.

$$BRPW_c = (H(r_i) \oplus pw_c)\|n \tag{1}$$

where the hash function $H(r_i)$ is used to transform the input r_i into a fixed-size string of bytes to represent the input data in a secure manner. The XOR $\oplus$ operation takes two inputs and produces a binary output where the bits are set to the value "1" if the corresponding inputs are different and to the value "0" if they are the same. The purpose of the arbitrary number n is to add randomness, making it harder to predict or reuse intercepted data.

Subsequently, the long-term key DS is deployed for encrypting id_c, as shown in Equation (2):

$$did_c = En_{DS}(id_c) \tag{2}$$

where En_{DS} is the encryption process, which is performed using a hybrid ECC–AES model with an SI-AO optimization. Here, SI-AO optimization is used for the key generation process, and the user id, id_c, is encrypted. Further, V_c transmits $\{did_c, BRPW_c\}$ to s via a communication medium. After obtaining the information from V, using a private key, K, the information is decrypted as did_c, attains an id_c value, and is further computed as shown in Equations (3)–(7) [29]:

$$id_c = dec_{DS}(did_c) \tag{3}$$

$$\beta_c = H(id_c \oplus b)\|a \tag{4}$$

$$\alpha_c = \beta_c \oplus H(id_c \oplus BRPW_c) \tag{5}$$

$$\delta_c = x_c \oplus H(\beta_c \oplus BRPW_c) \tag{6}$$

$$\lambda_c = H(id_c\|BRPW_c\|x_c\|\beta_c\ ) \tag{7}$$

Finally, the computed constraints $\{\alpha_c, \delta_c, \lambda_c, did_c, H(.)\}$ are accumulated in the smart card, and s transmits the smart card to V via a secured channel. Then V computes ρ_c following the reception message, as shown in Equation (8):

$$\rho_c = r_i \oplus n \oplus H(id_c \oplus pw_c) \tag{8}$$

Subsequently, ρ_c is accumulates on the smart card, and the registration procedure for the user is terminated. In this phase, encryption takes place using the hybrid ECC–AES model with an optimization strategy, where key generation is optimally accomplished with SI-AO.

The hybrid ECC–AES model for encryption is described as follows: A cubic non-singular curve with a sensible point in two parameters $f(p,q) = 0$ is known as EC [17] over an area M (i.e., an infinity point). Algebraic expansions of rationales, complex integers, m -adic integers, finite fields, and rationales are usually deployed as M. The basic field F_m of the EC group for cryptography is examined.

$$q^2 = p^3 + zp + o \tag{9}$$

Here, $m > 3$ denotes prime. An EC is a plane curve, as portrayed in Equation (5). Consider EC as shown in Equation (6).

$$G : q^2 = p^3 - p + 1 \tag{10}$$

If Z_1 and Z_2 are on G, which describes $Z_3 = Z_1 + Z_2$, assume $Z_1 = (p_1, q_1)$, $Z_2 = (p_2, q_2)$, $Z_3 = (p_3, q_3)$ and $Z_1 \neq Z_2$.

$$l = \frac{q_2 - q_1}{p_2 - p_1} \tag{11}$$

To find the meeting point with G, Equations (12)–(15) are followed:

$$(l(p - p_1) + q_1)^2 = p^3 + Kp + L \tag{12}$$

$$\text{Or, } 0 = p^3 - l^2 p^2 + \ldots\ldots \tag{13}$$

$$\text{So, } p_3 = l^2 - p_1 - p_2 \tag{14}$$

$$q_3 = l(p_1 - p_2) - p_1 \tag{15}$$

Multiplication is recognized as recurring addition; for example, $3Z = Z + Z + Z$. In the ECC cryptosystem [17], private and public keys are provided for every user. In addition, the public key is deployed for verifying and encrypting signatures. For decryption and creating signatures, a private key is exploited. Here, hybrid encryption keys are used for encrypting the text. At first, ECC-oriented encryption is conducted.

Furthermore, the encrypted text obtained from ECC is encrypted by means of AES encryption. The AES-based encryption model includes four transformations that rapidly disturb the plain text for enhanced security. It could, moreover, be implemented without difficulty in any paradigm due to its lower costs. AES has a predetermined block size of 128 bits and key sizes of 128, 192, and 256 bits, which have related cycle counts of 10, 12, and 14, respectively. It also encompasses four types of transformations, i.e., "Sub Bytes, Shift Rows, Mix Columns, and Add Round Key" [30]. Thus, AES-based encrypted text is generated. The private keys generated by AES are optimally chosen by means of SI-AO-based optimization.

3.3. Login Phase

The login phase is always initiated from the client side by sending a login request to the server. This requires several parameters, such as username and password.

1. The user V enters their own cth user identity idc', cth user password pwc', and bio info ri'.

2. After providing the info, the following factors are computed, as shown in Equations (16)–(20):

$$n = \rho_c \oplus r_i \oplus H\left(pw_c \oplus id_c{}'\right) \tag{16}$$

$$BRPW'_c = \left(H(r_i) \oplus pw'_c\right) \| n \tag{17}$$

$$\beta'_c = \alpha_c \oplus H\left(id'_c \oplus BRPW'_c\right) \tag{18}$$

$$x'_c = \delta_c \oplus H\left(\beta'_c \oplus BRPW'_c\right) \tag{19}$$

$$\lambda'_c = H\left(id'_c \| BRPW'_c \| x'_c \| \beta'_c\ \right) \tag{20}$$

Subsequently, λ'_c and λ_c are confirmed as equivalent. If they are equivalent, the confirmation passes; otherwise, the login request transmitted by V to s is discarded.

3. If the substantiation passes, the reader computes the following factors, as shown in Equations (21) and (22):

$$\omega_c = x'_c \oplus H\left(id'_c \oplus \beta'_c\right) \oplus H\left(id'_c \oplus \beta'_c \oplus T_1\right) \tag{21}$$

$$v_c = H\left(id'_c \| \beta'_c \| x'_c \ \| \left(\beta'_c \oplus x'_c\right) \| T_1\right) \tag{22}$$

After that, the login request $\{did_c, \omega_c, v_c, T_1\}$ is transmitted to the server.

3.4. Authentication Phase

This phase portrays the procedure of mutual substantiation among s and V. Following the transmission of login requests from the user to the server, the server begins to validate if V is legal by computing a series of constraints, and V confirms the legitimacy of s by computing the values of certain constraints. The procedure for authentication is portrayed comprehensively below.

1. Following the reception of the request by s from V, it initially confirms if the current timestamp is sensible. Further, *didc* is decrypted to attain *idc* and compute the following factors, as shown in Equation (23):

$$\begin{aligned} &\beta'_c = H\left(\left(id'_c \oplus b\right) \| a\right) \\ &x'_c = \omega'_c \oplus H\left(id'_c \oplus \beta'_c\right) \oplus \left(id'_c \oplus \beta'_c \oplus T_1\right) \\ &v'_c = H\left(id'_c \| \beta'_c \| x'_c \ \| \left(\beta'_c \oplus x'_c\right) \| T_1\right) \end{aligned} \tag{23}$$

s ensures if v'_c and v_c are equivalent. If not, s discards the login request from V. If equivalent, s obtains the login request from V and then evaluates the session key of two sides.

$$sk = H\left(id'_c \oplus \beta'_c \oplus x'_c \oplus T_1 \oplus T_2\right) \tag{24}$$

2. After computing the session key, s calculates Equation (25):

$$\varphi c = Hid'cx'c\beta'c \oplus x'cT2 \tag{25}$$

Subsequently, s transmits $\{\varphi_c, T_2\}$ to V.

3. Following reception of the message from s, the user initially confirms the legality of the time stamp T_2, and Equation (26) is computed:

$$\varphi'_c = H\left(id_c \| x'_c \| \left(\beta'_c \oplus x'_c\right) \| T_2\right) \tag{26}$$

V determines if φ'_c is equivalent to φ_c. If it is equivalent, V computes the session key, as shown in Equation (27):

$$sk = H\left(id_c \oplus \beta'_c \oplus x'_c \oplus T_1 \oplus T_2\right) \tag{27}$$

Thus, the authentication procedure for V and s is finished.

Information flow between V_1 and V_2 is described below.

From V_1 (virtual computer), the data file is transmitted to V_2 (virtual machine). When the criterion is satisfied, the entity of V_1, with the information, flows to the entity V_2. This is given by Equation (28):

$$V_1 \to bifid_{V_1} - Ability^{-}_{V_1} \subseteq id_{V_2} + Ability^{-}_{V_2} \tag{28}$$

When data is received by the virtual machine, the security mark of V_2 is interrupted, and the required alterations are conducted. This is shown in Equation (29):

$$id'_{V_2} \leftarrow id_{V_2} \cup \left(id_{V_1} - Ability^{-}_{V_1}\right) \tag{29}$$

The low-security information flows are protected to safeguard the domain's higher security level by considering the transmitting ability of V_1 and the reception ability of the information inflow entity V_2, as well as the execution of identity.

3.5. Decryption

Finally, the ECC-oriented model is applied to decrypt the encrypted data. After that, the attained ECC-based decrypted data is further decrypted by means of the AES approach.

4. SI-AO-Based Optimization for Optimal Key Selection Objective:

The objective Obj is to minimize the correlation between original data and encrypted data, as shown in Equation (30), where $Corr$ refers to correlation:

$$Obj = min(Corr) \tag{30}$$

4.1. Solution Encoding

In this work, the private keys, denoted as K, generated by AES, are optimally chosen by means of SI-AO-based optimization. Figure 2 shows the representation of solutions, wherein wn stands for the entire count of private keys.

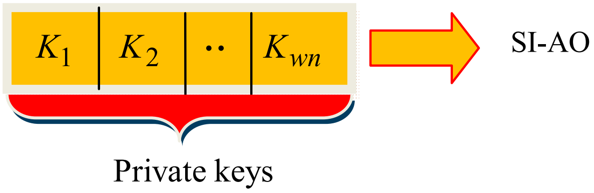


Figure 2. Solution encoding.

While the Aquila Optimizer (AO) [31] retains strong exploration ability, it lacks exploitation ability. For this reason, the conservative process required specific adjustments. Overall, improved convergence can be reached through self-enhancement due to its competency in conservative optimization models [32–35].

4.2. Initialization

Aquila is a population-based algorithm; hence, the tuning begins with the population of the potential solution. At this point, the best-obtained solution is evaluated and established. In this work, chaotic opposition-based learning (C-OBL) is carried out, and opposite solutions $\left(\hat{P}_1\right)$ are created. Based on Equation (31), solutions $A_{i,j}$ denote the current candidate set, which is constructed randomly. A_i represents the i^{th} solution's position; S is the solution count indicating the total number of solutions in the candidate set, essentially denoting the size of the population being considered in the Aquila algorithm; and Dim refers to the dimensional size.

$$A_{i,j} = ran \times (ub_k - lb_k),\ i = 1,2,\ldots Sj = 1,2,\ldots Dim \tag{31}$$

Here, *ran* refers to a randomized number, and lb_k and ub_k are the lower and upper bounds.

4.3. Mathematical Model

The proposed SI-AO algorithm, which is based on the hunting behavior of Aquila and the intelligence behavior of the bird's swarm, is explained in the following sub-sections:

- Step I: Extended exploration (P_1):

Aquila discovers and votes for the finest hunting region through higher soar using perpendicular scoop topology, which can be mathematically defined by Equation (32):

$$P_1(t^\gamma{+}1) = \mathrm{P_{best}}(t^\gamma) \times (1{-}\frac{t^\gamma}{T}) + (P_M(t^\gamma) - P_{best}(t^\gamma) \times ran) \tag{32}$$

$$P_M(t) = \frac{1}{S}\sum_{i=1}^{S} P_i(t^\gamma) \tag{33}$$

$P_{\mathrm{best}}(t^\gamma)$ is the finest location as yet attained;
$P_M(t^\gamma)$ is the total Aquila's average position in the current iteration;
t^γ is the present iteration;
T is the total count of iterations;
S is the population size;
a_1 is a randomized number between (0, 1).

- Step II: Narrow exploration (P_2):

During the constricted exploration, Aquila employs the technique identified as "The contour flight with short glide attack, which is also called narrowed exploration". In this technique, a short glide is adopted inside the selected area to attack the prey. For this reason, the location is updated as shown in Equation (34):

$$P_2(t^\gamma + 1) = P_{best}(t^\gamma) \times Levy(N) + P_R(t^\gamma) + (l - m) \times ran \tag{34}$$

Here, $P_R(t)$ is Aquila's random position;
$Levy(N)$ is the levy flight distribution function.
Here, t = 001, β = 1.5, u, y represents an arbitrary integer, and σ is computed as per Equation (35):

$$\begin{cases} l = a \times sin(\theta) \\ m = a \times cos(\theta) \\ a = a_3 + 0.00565 \\ \theta = -\omega \times N_1 + \frac{3\times\pi}{2} \end{cases} \tag{35}$$

where a_3 is the number of search cycles among 1 and 20, N_1 is the integer numbers within 1 and N, and ω = 0.005.

- Step III: Extended exploitation (P_3):

Aquila's preliminary attack occurs through vertical descent. To be precise, Aquila gradually descends until it approaches the target. This method is referred to as extended exploitation; during this mode, prey areas are roughly discovered. This behavior is modeled in Equation (36):

$$P_3(t^\gamma + 1) = (P_{best}(t) - p_M(t^\gamma)) \times \alpha' - ran + ((ub - lb) \times ran + lb) \times \delta' \tag{36}$$

Here, the parameters α' and δ' are the exploitation adjustment parameters, which are fixed (0.1).

$P_M(t)$ is the current positions' mean value and *ran* denotes a randomized number between (0, 1).

- Step IV: Narrowed exploitation (P_4):

As Aquila approaches the prey, the prey is attacked according to stochastic movements, as shown in Equation (40), where qf indicates the quality factor, V_1 refers to AO motion variety, and V_2 implies AO's flight slope that decreases from 2 to 0. Conventionally, V_2 is computed as in Equation (37). As per the developed SI-AO approach, V_2 is computed using Equation (38), wherein ϕ is evaluated as in Equation (37):

$$V_2 = 2 * \left(1 - \frac{1}{T}\right) \tag{37}$$

$$V_2 = \phi(2 * ran - 1) \tag{38}$$

$$\phi = 2 * \left(1 - \left(\frac{t}{T}\right)^{\frac{1}{3}}\right)^{\frac{1}{3}} \tag{39}$$

$$P_4(t^\gamma + 1) = qf \times P_{best}(t) - (V_1 \times P(t^\gamma) \times ran) - V_2 \times levy(N) + ran \times V_1 \tag{40}$$

Pseudocode of SI-AO is shown in Algorithm 1.

Algorithm 1: SI-AO model

```
Initialize the population $P$;
Create C-OBL-based solutions;
Set the parameters;
while (condition $\neq$ end) do
    Compute fitness (Fit);
    $P_{best}(t^\gamma)$=Find best solution as per Fit;
    for (i=1,2...,M) do
        Upgrade solution average value $P_M(t^\gamma)$;
        Upgrade $l, m, V_1, V_2, levy(N)$;
        if $t \leq (2/3) * T$ then
            if $rand \leq 0.5$ then
                Step 1: Upgrade solution (Eq. 32);
                if $Fit(p_1(t^\gamma + 1)) < Fit(p(t^\gamma))$ then
                    $p_M(t^\gamma) = (p_1(t^\gamma + 1))$;
                end
                if $Fit(p_1(t^\gamma + 1)) < Fit(X_{best}(t))$ then
                    $P_{best}(t^\gamma) = P_1(t^\gamma + 1)$;
                end
            end
            else
                Step 2: Upgrade solution (Eq.34);
                if $Fit(P_2(t^\gamma + 1)) < Fit(P(t^\gamma))$ then
                    $P(t^\gamma) = (P_2(t^\gamma + 1))$;
                end
                if $Fit(X_2(t + 1)) < Fit(X_{best}(t))$ then
                    $P_{best}(t^\gamma) = P_2(t^\gamma + 1)$
                end
                else
                    if $rand \leq 0.5$ then
                        Step 3: Upgrade solution (Eq.36);
                        if $Fit(P_2(t^\gamma + 1)) < Fit(P(t^\gamma))$ then
                            $P(t^\gamma) = (P_3(t^\gamma + 1))$;
                            if $Fit(X_3(t + 1)) < Fit(X_{best}(t))$ then
                                $P_{best}(t^\gamma) = P_3(t^\gamma + 1)$;
                            end
                        end
                        else
                            Compute $V_2$ as in (Eq.38);
                            Step 4: Upgrade solution (Eq.40);
                            if $Fit(P_4(t^\gamma + 1) < Fit(P(t^\gamma))$ then
                                $P(t^\gamma) = P_4(t^\gamma + 1)$;
                                if $Fit(P_4(t^\gamma + 1)) < Fit(P_{best}(t^\gamma))$ then
                                    $P_{best}(t^\gamma) = P_4(t^\gamma + 1)$;
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
return $X_{best}$
```

5. Results and Discussions

5.1. Simulation Procedure

The presented SI-AO scheme for the proposed authentication protocol was executed in MATLAB (Version 3.0.0) with ThingSpeak. This allows for a controlled and flexible setting to test and refine the protocol. In this setting, computations are typically executed on a general-purpose CPU, which benefits from different optimization strategies compared to the embedded processors commonly used in IoT devices. This may affect the performance metrics, such as execution time, energy consumption, and memory usage. Accordingly, an investigation was made by means of the manual dataset. The performance of the developed scheme, as measured by diverse metrics, was compared to existing models, such as the lion algorithm (LA) [36], the butterfly optimization algorithm (BOA) [37], the spider monkey optimization (SMO) [38], the Aquila Optimizer (AO) [32], poor and rich optimization (PRO) [39], the distant user authenticated protocol [22], the secure and lightweight authentication scheme [25], blowfish, and Rivest–Shamir–Adleman (RSA). Here, the convergence analysis was conducted for iterations such as 0, 5, 10, 15, 20, and 25. Furthermore, an examination was held related to diverse attacks such as the chosen-plaintext attack (CPA) and the chosen-ciphertext attack (CCA). “A CPA is an attack model for cryptanalysis that presumes that the attacker can obtain the ciphertexts for arbitrary plaintexts. A CCA is an attack model for cryptanalysis where the cryptanalyst can gather information by obtaining the decryptions of chosen ciphertexts. From these pieces of information, the adversary can attempt to recover the hidden secret key used for decryption. The goal of the attack is to gain information that reduces the security of the encryption scheme”. Moreover, the time efficiency (decryption time and encryption time) and cost efficiency of the presented method were analyzed.

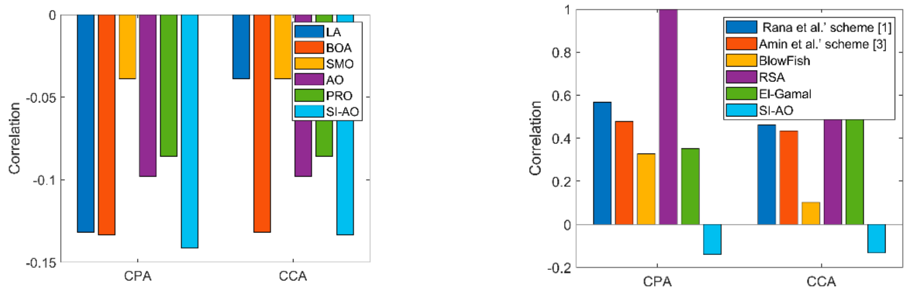
5.2. Simulation Platform

The implementation of the SI-AO scheme for the authentication protocol was implemented using MATLAB integrated with ThingSpeak. This setup presents a regulated and adapTable setting for protocol testing and improvement. MATLAB, as a high-level language, provides an interactive environment with features for data analysis, algorithm development, numerical computation, and visualization. It supports various toolboxes and functions that make it simpler to model sophisticated systems and carry out complex mathematical operations. ThingSpeak is a cloud-based IoT analytics platform that allows real-time data collection, analysis, and visualization, designed for IoT applications. The platform has the ability to display and evaluate data. This makes it feasible for protocol testing in virtual IoT environments. MATLAB’s capabilities are enhanced by the integration of ThingSpeak, enabling data transfer between them. This allows continuous monitoring and analysis of IoT applications.

By leveraging MATLAB’s computational power and ThingSpeak data handling capabilities, the SI-AO model authentication protocol can be evaluated, as this provides a comprehensive and versatile environment for evaluation under various performance conditions, ensuring the required standards of efficiency and security for IoT applications are met.

5.3. Attack Analysis

This section describes the attack analysis of the developed SI-AO model over existing approaches. Here, the evaluation was conducted using a manual dataset, and the related outcomes are shown in Figure 3. The manual dataset was curated to cover a wide range of scenarios that the SI-AO model could potentially encounter. Accordingly, this ensured that the model was tested under numerous conditions. In addition, the dataset was used to further evaluate both CPA and CCA, involving different modes of cryptanalysis. Figure 3a describes the attack analysis of the developed SI-AO model over extant optimization schemes (LA, BOA, SMO, AO, and PRO) regarding diverse attacks such as CPA and CCA, as these are among the most prevalent attacks in cryptanalysis. CPA tests the system’s resilience, while CCA evaluates the system's vulnerability. Therefore, analyzing these attack models potentially ensures a comprehensive analysis of the SI-AO model's security.



(**a**) (**b**)

Figure 3. Attack analysis using SI−AO over extant (**a**) optimization schemes and (**b**) cryptographic schemes regarding CPA and CCA.

A CPA is an attack model for cryptanalysis that presumes the attacker can obtain the ciphertexts for arbitrary plaintexts. A chosen-ciphertext attack (CCA) is an attack model for cryptanalysis where the cryptanalyst can gather information by obtaining the decryptions of chosen ciphertexts. Figure 3b describes the CPA and CCA attack analysis of the developed SI-AO model over extant cryptographic schemes (distant user authenticated protocol [20], secure and lightweight authentication scheme [25], blowfish, RSA, and ElGamal). The objective of this work was to minimize the correlation between original data and encrypted data and thereby prevent the hacker from extracting the original data from the network. Thus, if the objective considered in Equation (21) was fulfilled, then the attack rates would be negligible. For both cryptographic comparison and optimization comparison, the introduced SI-AO scheme achieved better outcomes than the compared schemes. Primarily, slight CPA and CCA values assured the minimal attack rates of the SI-AO scheme. As seen in Figure 3, the deployed SI-AO scheme accomplished the least CPA attack rate of −0.135, while the other evaluated schemes, LA, BOA, SMO, AO, and PRO, accomplished relatively higher attack rate values of −0.128, −0.13, −0.04, −0.09, and −0.08, respectively. The attack rates were quantitatively measured and compared. The SI-AO model achieved the lowest CPA attack rate of −0.135. This is considered a superior performance in minimizing the correlation between the original and encrypted data, and it can be concluded that the correlation between the original data and retrieved data was low. Consequently, the analysis proved the superior efficacy of SI-AO with its optimized encryption theory.

5.4. Convergence Analysis

The convergence (cost) analysis of the presented SI-AO method over traditional schemes (LA, BOA, SMO, AO, and PRO) for different iterations is illustrated in Figure 4. The evaluation was conducted by adjusting the iterations from 0 to 5, 10, 15, 20, and 25. Lower cost values would indicate improved system performance. The presented SI-AO attained the minimum cost values, ranging from the 12th iteration to the 25th iteration. Moreover, from iteration 0 to 12, the cost values were somewhat higher for the developed model, whereas from iteration 13 to iteration 25, the proposed model showed comparatively lower cost values. At the final iterations (13–25), minimal cost values (−0.189) were attained. Here, extant SMO and PRO models revealed the worst performance over LA, BOA, and AO schemes. Therefore, the overall assessment corroborates the better performance of the developed model with the inclusion of the introduced SI-AO concept.

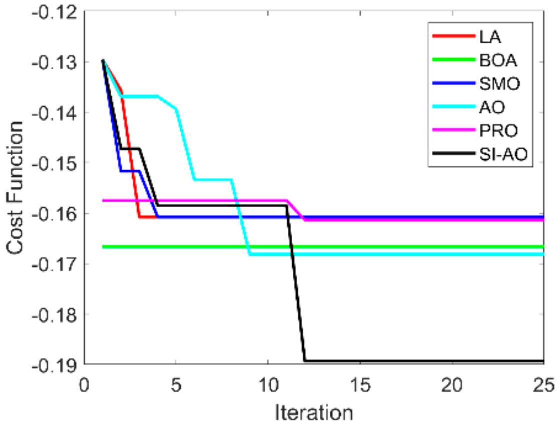


Figure 4. Convergence analysis of the SI−AO approach over compared approaches.

5.5. Analysis of Encryption Time and Decryption Time

Table 2 shows the decryption and encryption times of the developed SI-AO scheme compared to the existing schemes LA, BOA, SMO, AO, PRO, distant user authenticated protocol, secure and lightweight authentication scheme [25], blowfish, RSA, and ElGamal. In fact, encryption time and decryption time must be minimal for the system's improved performance. It can be seen that SI-AO achieved the minimum values compared to the existing schemes, meaning it took less time to encrypt and decrypt the messages. Moreover, the time taken by the developed model to decrypt the message, 0.0225 s, was greater than the time taken to encrypt the message, 0.0142 s. These results demonstrate the enhancements attained due to the newly developed concepts in the proposed SI-AO model.

Table 2. Analysis of encryption and decryption times.

Methods	Encryption Time	Decryption Time
Distant user authenticated protocol [20]	0.0267	0.0226
Secure and lightweight authentication scheme [25]	0.0224	0.0256
BlowFish	1.2502	1.0388
RSA	0.10865	0.10886
ElGamal	0.016656	0.032491
LA	0.018907	0.025421
BOA	0.016194	0.037711
SMO	0.01574	0.024191
AO	0.015745	0.024057
PRO	0.015473	0.023785
SI-AO	0.014281	0.022525

5.6. Analysis of Computation Time and Computation Cost

Table 3 shows the analysis regarding computation time, and Table 4 shows the analysis of computation cost. These aspects were analyzed to prove the enhanced performance of the employed SI-AO approach in contrast to traditional methods. The conservative schemes considered here were LA, BOA, SMO, AO, and PRO. Computation time must be minimal for superior system performance. The proposed SI-AO scheme achieved negligible improvement over the LA, BOA, SMO, AO, and PRO methods. Next to SI-AO, the AO model had the lowest computation time values compared to the LA, BOA, SMO, and PRO schemes. Similarly, the developed SI-AO scheme gained smaller computation cost values over the LA, BOA, SMO, AO, and PRO methods. Namely, a minimal computation cost value of −0.18931 was gained by the SI-AO model, which was negligible compared to the values gained by the LA, BOA, SMO, AO, and PRO methods. Thus, the enhancement of the adopted scheme was proven.

Table 3. Analysis of computation time.

Methods	Computation Time
LA	32.051
BOA	33.415
SMO	20.795
AO	20.676
PRO	62.465
SI-AO	20.012

Table 4. Analysis of computation costs.

Methods	Computation Cost
LA	−0.1608
BOA	−0.16664
SMO	−0.1608
AO	−0.16813
PRO	−0.16138
SI-AO	−0.18931

5.7. Statistical Analysis

The statistical analysis of the proposed SI-AO method was computed compared to the traditional models for fitness function and key sensitivity, and the results are depicted in Tables 5 and 6. As seen in Table 5, the best-case scenario showed an improvement in the proposed SI-AO model, which was −17.73%, −13.60%, −17.73%, −12.60%, and −17.31% better than the traditional models LA, BOA, SMO, AO, and PRO, respectively. The mean performance of the adopted SI-AO approach for key sensitivity showed better results than the traditional schemes. As a result, the improvement in the proposed SI-AO model has been validated effectively in all cases.

Table 5. Statistical analysis of fitness function.

Methods	Best	Worst	Mean	Median	Std
LA	−0.1608	−0.12961	−0.15855	−0.1608	0.0078522
BOA	−0.16664	−0.16664	−0.16664	−0.16664	2.83E−17
SMO	−0.1608	−0.12961	−0.15883	−0.1608	0.0065853
AO	−0.16813	−0.12961	−0.15994	−0.16813	0.013225
PRO	−0.16138	−0.15749	−0.15967	−0.16138	0.0019707
SI-AO	−0.18931	−0.12961	−0.17371	−0.18931	0.018922

Table 6. Statistical analysis of key sensitivity.

Methods	Best	Worst	Mean	Median	Std
LA	−0.13338	0.20836	−0.034569	−0.080333	0.1379
BOA	−0.096264	0.16962	−0.025719	−0.08559	0.1127
SMO	−0.13338	0.20836	−0.041368	−0.080333	0.1425
AO	−0.08886	0.1806	0.026558	−0.054052	0.1413
PRO	−0.11499	−0.010926	−0.069182	−0.063782	0.041
SI-AO	−0.13724	−0.096563	−0.12124	−0.12221	0.0156

5.8. Friedman Test

“The Friedman test is a non-parametric statistical test, similar to the parametric repeated measures ANOVA, it is used to detect differences in treatments across multiple test attempts”. Table 7 shows that the proposed SI-AO model accomplished better performance by meeting the fitness function with the minimum rank of 2.1429, whereas the existing models such as LA, BOA, SMO, AO, and PRO accomplished a higher rank.

Table 7. Friedman test of the proposed model over traditional methods.

Methods	Rank
p-value	1.698×10^{-8}
Sigma	1.8439
LA	4.6429
BOA	2.4762
SMO	4.6429
AO	2.619
PRO	4.4762
SI-AO	2.1429

5.9. Analysis of Wilcoxon Signed-Rank Test

"The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test used either to test the location of a population based on a sample of data or to compare the locations of two populations using two matched samples". Table 8 shows that the proposed SI-AO model had better value when compared to conventional models such as LA, BOA, SMO, AO, and PRO. Thus, the proposed SI-AO scheme represents an improvement over previous models.

Table 8. Wilcoxon signed-rank test results.

Methods	LA	BOA	SMO	AO	PRO	SI-AO
Probability	1.31×10^{-6}	5.73×10^{-7}	1.83×10^{-6}	5.49×10^{-6}	6.41×10^{-6}	7.35×10^{-6}
Normal (Z) statistic	−4.8378	−5	−4.7717	−4.5451	−4.5124	−4.4833

5.10. Analysis of Brute Force Attack and Man-in-the-Middle Attack

The outcomes of the brute force attack and man-in-the-middle attack are provided in Table 9. The results confirm that the proposed SI-AO scheme achieved better results than comparable schemes. The proposed SI-AO scheme attained a higher value of 0.0015732 for the brute force attack, which was 15.01%, 14.06%, 33.78%, 72.92%, and 19.00% greater than that for LA, BOA, SMO, AO, and PRO, respectively.

Table 9. Analysis of brute force attacks and man-in-the-middle attacks.

Attacks	LA	BOA	SMO	AO	PRO	SI-AO
Brute force Attack	0.001337	0.001352	0.0010417	0.0004261	0.0012743	0.0015732
Man-in-the-Middle Attack	0.60239	0.50026	0.11024	0.6581	0.40313	0.70112

6. Conclusions

A novel authentication protocol was introduced for IoT, where the initial user registration phase involved encryption with a hybrid ECC–AES model with an optimization strategy, whereby key generation was optimally achieved via SI-AO. The second and third phases included the login process and the authentication phase, in which information flow control-based authentication was conducted. Finally, decryption was achieved based on the hybrid ECC–AES model. Further analysis showed that the developed SI-AO scheme gained negligible values over the LA, BOA, SMO, AO, and PRO methods. The AO model achieved lower computation time values than the LA, BOA, SMO, and PRO schemes. Similarly, the developed SI-AO scheme gained smaller computation cost values than the LA, BOA, SMO, AO, and PRO methods. A minimal computation cost value of −0.18931 was gained by the SI-AO model, which was negligible compared to that gained by the LA, BOA, SMO, AO, and PRO methods. In the future, the new system will be practically implemented for better performance on real low-end embedded IoT platforms, and traceability attack analysis will be performed. Moreover, energy consumption based on PoC implementation may also be considered.

Author Contributions: Conceptualization, A.M. and B.A.; Formal analysis, A.M. and B.A.; Methodology, B.A.; Project administration, B.A. Writing—original draft, A.M. and B.A.; Writing—review and editing, A.M. and B.A. All authors will be informed about each step of the manuscript processing, including submission, revision, revision reminder, etc., via emails from our system or the assigned Assistant Editor. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

Abbreviations	**Full form**
IoT	Internet of things
AES	Advanced Encryption Standard
ECC	Elliptic Curve Cryptography
SI-AO	Self-Improved Aquila Optimizer
PUF	Physical Unclonable Functions
ILAS-IoT	Lightweight Authentication Scheme for IoT Deployments
ESL	Ephemeral Secret Leakage
BAN	Barrows-Abadi-Needham
ROR	Real-or-Random
M2M	Machine to Machine
AO	Aquila Optimizer
C-OBL	Chaotic Opposition Based Learning
LA	Lion Algorithm
BOA	Butterfly Optimization Algorithm
SMO	Spider Monkey Optimization
PRO	Poor and Rich Optimization
RSA	Rivest-Shamir-Adleman
CPA	Chosen-Plaintext Attack
CCA	Chosen-Ciphertext Attack

References

1. Airehrour, D.; Gutierrez, J.A.; Ray, S.K. SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things. *Future Gener. Comput. Syst.* **2019**, *93*, 860–876. [CrossRef]
2. Conti, M.; Kaliyar, P.; Rabbani, M.M.; Ranise, S. Attestation-enabled secure and scalable routing protocol for IoT networks. *Ad Hoc Netw.* **2020**, *98*, 102054. [CrossRef]
3. Deebak, B.D.; Al-Turjman, F. A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks. *Ad Hoc Netw.* **2020**, *97*, 102022.
4. Liu, L.; Ma, Z.; Meng, W. Detection of multiple-mix-attack malicious nodes using perceptron-based trust in IoT networks. *Future Gener. Comput. Syst.* **2019**, *101*, 865–879. [CrossRef]
5. Rathore, S.; Park, J.H. Semi-supervised learning based distributed attack detection framework for IoT. *Appl. Soft Comput.* **2018**, *72*, 79–89. [CrossRef]
6. Awan, K.A.; Din, I.U.; Zareei, M.; Talha, M.; Guizani, M.; Jadoon, S.U. HoliTrust-A holistic cross-domain trust management mechanism for service-centric Internet of Things. *IEEE Access* **2019**, *7*, 52191–52201. [CrossRef]
7. Bu, L.; Isakov, M.; Kinsy, M.A. A secure and robust scheme for sharing confidential information in IoT systems. *Ad Hoc Netw.* **2019**, *92*, 101762. [CrossRef]
8. Yaser, A.; Alsahlani, F.; Popa, A. LMAAS-IoT: Lightweight multi-factor authentication and authorization scheme for real-time data access in IoT cloud-based environment. *J. Netw. Comput. Appl.* **2021**, *192*, 103177.
9. Alotaibi, M. Security to wireless sensor networks against malicious attacks using Hamming residue method. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 8. [CrossRef]

10. Raoof, A.; Matrawy, A.; Lung, C. Routing attacks and mitigation methods for RPL-based Internet of Things. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1582–1606. [CrossRef]
11. Xu, T.; Gao, D.; Dong, P.; Zhang, H.; Foh, C.H.; Chao, H.-C. Defending against new-flow attack in SDN-based Internet of Things. *IEEE J. Mag.* **2017**, *5*, 3431–3443. [CrossRef]
12. Schweitzer, N.; Stulman, A.; Margalit, R.D.; Shabtai, A. Contradiction based gray-hole attack minimization for Ad-Hoc networks. *IEEE J. Mag.* **2017**, *16*, 2174–2183. [CrossRef]
13. Lomotey, R.K.; Pry, J.; Sriramoju, S. Wearable IoT data stream traceability in a distributed health information system. *Pervasive Mob. Comput.* **2017**, *40*, 692–707. [CrossRef]
14. Zhou, B.; Zhang, Q.; Shi, Q.; Yang, Q.; Yu, Y. Measuring web service security in the era of Internet of Things. *Comput. Electr. Eng.* **2017**, *66*, 305–315. [CrossRef]
15. Moosavi, S.R.; Gia, T.N.; Rahmani, A.M.; Nigussie, E.; Tenhunen, H. SEA: A secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways. *Procedia Comput. Sci.* **2015**, *52*, 452–459. [CrossRef]
16. Sciancalepore, S.; Piro, G.; Vogli, E.; Boggia, G.; Cavone, G. LICITUS: A lightweight and standard compatible framework for securing layer-2 communications in the IoT. *Comput. Netw.* **2016**, *108*, 66–77. [CrossRef]
17. Gampala, V.; Inuganti, S.; Muppidi, S. Data security in cloud computing with elliptic curve cryptography. *Int. J. Soft Comput. Eng. IJSCE* **2012**, *2*, 1–14.
18. Memos, V.A.; Psannis, K.E.; Ishibashi, Y.; Kim, B.G.; Gupta, B.B. An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework. *Future Gener. Comput. Syst.* **2017**, *83*, 619–628. [CrossRef]
19. Tedeschi, S.; Mehnen, J.; Tapoglou, N.; Roy, R. Secure IoT devices for the maintenance of machine tools. *Procedia CIRP* **2017**, *59*, 150–155. [CrossRef]
20. Rana, M.; Shafiq, A.; Altaf, I.; Alazab, M.; Mahmood, K.; Chaudhry, S.A.; Bin Zikria, Y. A secure and lightweight authentication scheme for next generation IoT infrastructure. *Comput. Commun.* **2021**, *165*, 85–96. [CrossRef]
21. Melki, R.; Noura, H.N.; Chehab, A. Lightweight multi-factor mutual authentication protocol for IoT devices. *Int. J. Inf. Secur.* **2020**, *19*, 679–694. [CrossRef]
22. Fotouhi, M.; Bayat, M.; Doostari, M.A. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT. *Comput. Netw.* **2020**, *177*, 107333. [CrossRef]
23. Alzahrani, B.A.; Chaudhry, S.A.; Barnawi, A. ILAS-IoT: An improved and lightweight authentication scheme for IoT deployment. *J. Ambient. Intell. Hum. Comput.* **2020**, *13*, 5123–5135. [CrossRef]
24. Khalid, U.; Asim, M.; Baker, T. A decentralized lightweight blockchain-based authentication mechanism for IoT systems. *Clust. Comput.* **2020**, *23*, 2067–2087. [CrossRef]
25. Das, A.K.; Bera, B.; Wazid, M.; Jamal, S.S.; Park, Y. On the security of a secure and lightweight authentication scheme for next generation IoT infrastructure. *IEEE Access* **2021**, *9*, 71856–71867. [CrossRef]
26. Jebri, S.; Amor, A.B.; Abid, M. Enhanced lightweight algorithm to secure data transmission in IoT systems. *Wirel. Pers. Commun.* **2021**, *116*, 2321–2344. [CrossRef]
27. Seunghwan, S.; Park, Y.; Park, Y. A Secure, Lightweight, and Anonymous User Authentication Protocol for IoT Environments. *Sustainability* **2021**, *13*, 9241. [CrossRef]
28. Ehui, B.B.; Han, Y.; Guo, H.; Liu, J. A Lightweight Mutual Authentication Protocol for IoT. *J. Commun. Inf. Netw.* **2022**, *7*, 181–191. [CrossRef]
29. Chen, C.M.; Liu, S. Improved secure and lightweight authentication scheme for next-generation IOT infrastructure. *Secur. Commun. Netw.* **2021**, *2021*, 6537678. [CrossRef]
30. Yin, A.; Wang, S. A novel encryption scheme based on timestamp in gigabit ethernet passive optical network using AES-128. *Optik* **2014**, *125*, 1361–1365. [CrossRef]
31. Abualigah, L.; Yousri, D.; Elaziz, M.A.; Ewees, A.A.; Mohammed, A.A.A.-Q.; Gandomi, A.H. Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
32. Mahajan, S.; Abualigah, L.; Pandit, A.K.; Altalhi, M. Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks. *Soft Comput.* **2022**, *26*, 4863–4881. [CrossRef]
33. Wagh, M.B.; Gomathi, N. Improved GWO-CS algorithm-based optimal routing strategy in VANET. *J. Netw. Commun. Syst.* **2019**, *2*, 34–42.
34. Halbhavi, B.S.; Kodad, S.F.; Ambekar, S.K.; Manjunath, D. Enhanced invasive weed optimization algorithm with chaos theory for weightage based combined economic emission dispatch. *J. Comput. Mech. Power Syst. Control.* **2019**, *2*, 19–27.
35. Jadhav, A.N.; Gomathi, N. DIGWO: Hybridization of dragonfly algorithm with improved grey wolf optimization algorithm for data clustering. *Multimed. Res.* **2019**, *2*, 1–11.
36. Boothalingam, R. Optimization using lion algorithm: A biological inspiration from lion's social behavior. *Evol. Intell.* **2018**, *11*, 31–52. [CrossRef]
37. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [CrossRef]

38. Harish, S.; Garima, H.; Jagdish, B. Spider monkey optimization algorithm. In *Evolutionary and Swarm Intelligence Algorithms*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 779, pp. 43–59.
39. Seyyed, M.; Vahid, B. Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Eng. Appl. Artif. Intell.* **2019**, *86*, 165–181.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.