Received: March 13, 2022. Revised: May 14, 2022. 

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

# Lightweight Authentication Model for IoT Environments Based on Enhanced Elliptic Curve Digital Signature and Shamir Secret Share

**Mohammed Shakir Oudah[1]*** **Abeer Tariq Maolood[1]**

[1] *Computer science department, university of technology, Baghdad, Iraq*
* Corresponding author's Email: cs.20.42@grad.uotechnology.edu.iq

**Abstract:** With the rapid use of the Internet, and the increase in the numbers of smart and mobile devices, the internet of things (IoT) fields had become more crucial fields in the interest of developers. Many aspects related to authentication and privacy communication during an interaction between remote users and IoT devices should be enabled and achieved in a lightweight and secure manner. Due to the elliptic curve digital signature (ECDSA)'s features that are secure and lightweight compared with other public key algorithms; this paper presents a new IoT authentication model that incorporates a modified ECDSA. But unfortunately, the familiar use of ECDSA in Blockchain showed some problems related to the capability of revealing the random private integer, which leads to private key disclosure, and hence funds theft, this paper highlights this problem and proposes modifications to the ECDSA to make it more reliable. In addition, the proposed model combines a modified ECDSA and Shamir's secret sharing (SSS). The combination can give better results in establishing a more securely authentication agreement and robustness to resolve the standard algorithm attenuation, this modification involved a variant of ECDSA's calculations in signature processing, and employing Shamir's Secret Sharing to further protect the random private integer. Also, the proposed model achieves lower overhead communications by splitting and redistributing authentication calculations roles between the system's entities. After security analysis, the proposed algorithm exhibits ability to resist potential threats, also, a comparison with other related works demonstrates that the proposed algorithm performs fewer arithmetic operations, relatively, by decreasing the number of modular inverse operations, so this feature lowers resources-efforts and high performance, moreover, the theoretical analysis indicates that the proposed algorithm efficiently manages the scalability of the system.

**Keywords:** IoT, Elliptic curve digital signature, Shamir's secret sharing.

## 1. Introduction

The internet of things (IoT) technology appears as a digital revolution in information technology, with huge numbers of devices that are connected and exchange their critical and real-time data, and perform their processes on them simultaneously. According to oracle, the number of interconnected IoT devices will grow to 22 billion by 2025 [1]. IoT devices can be deployed in different areas such as industrial IoT (IIoT), for instance, a smart factory that involved many physical devices with a centralized monitor system, these devices cooperate to perform some tasks that are requested by humans in real-time via wired or wireless communication systems [2]. Another IoT image appears as wireless sensor network (WSN), that do intelligent tasks such as collecting real-time data from surrounding environments and transmitting them to the remote requested hosts [3]. Also, the physical security equipment to monitor and control human access such as door locks, and other surveillance systems [4].

IoT devices run in an open network (Internet) and use their facilities to transmit data [5], so these data can be disclosed by different cyber-attacks [6], therefore, the most concerns involved how to establish efficient security models, and authentication mechanisms to prevent maliciously, and unauthorized access to IoT devices and stealing of critical transmission data that may be transmitted through an insecure channel. Noteworthy, the

authentication mechanisms should be chosen appropriately to manipulate many constraints such as the low capacity of IoT devices, the low energy, and the low processing power of them, so any mechanisms should be low complex, lower resources-consuming, and lightweight overall in their performance [7].

At first glance, this seems like a tradeoff between performance and security ability, therefore, the main challenge [8] in a limited-environment platform, such as IoT, is how to consider the appropriate features in any security mechanism. Many lightweight authentication methods are used for established better key agreements such as password authentication, biometrics authentication, digital signature algorithms, and so on. In this paper, enhancing an Elliptic curve digital signature algorithm (ECDSA) and exploiting it to achieve an IoT-authentication model, comes with two main reasons. Firstly, the ECDSA is a popular method that is used in many applications to authorize and identify a user's identity, but proper exploitation of standard ECDSA [9] should undergo some modifications and improvements to get rid of the probability of revealing the private key when repeat usage of the integer randomness selection, moreover, the proposed algorithm is a hybrid algorithm, employs Shamir's secret sharing algorithm, as another security level, to protect the selected random integer, so, the algorithm's scheme is focused on making the authentication agreement more reliable to resist the man-in-the-middle, and forger signature attacks and other spam transactions, whenever the random integer key is reused. secondly, the hybrid method provides a flexible way to perform its calculations with the steady expansion of systems, where the authentication model performance doesn't require more storage or processing power, to adapt within a limited-resources environment, such as IoT.

## 2. Related work

The authentication model that is applied to restrict the IoT device's access can take different approaches to verify the user's identity, these approaches are classified as [8], One-time-password authentication (OTP), ID-based authentication, and user's certificate authentication, and each class involved several researchers methods [2, 4, 9]. The motives of this paper, are to establish a reliable and lightweight authentication model, so emphasizing various research works that employed the ECC to introduce the IoT authentication model.

A comparative study in [10] involved showing how the security model performance in IoT environments can be improved by using ECC instead of RSA, this study was implemented upon evaluation of the smart-door IoT system using Cryptool2 as a simulation tool. Yasin Genç [11] proposed a new ECDSA algorithm that involved eliminating modular inversion arithmetic operation in two signature stages, the signature generation stage and verification stage, the elimination improves the run time of the ECDSA algorithm, and claimed the modification doesn't reduce the security efficiency. It seems that the [11] didn't address the random integer reuse problem in ECSDA, the modifications were implemented to increase the run time performance without taking into account the security weakness in the standard algorithm. Thus, the new version of the calculations is still suffered from the problem of reusing the random integer with the same private key.

Two authenticated key agreement (AKA) methods are proposed in [12], the first method is an implicit certificate was implemented using EEC, the main purpose of using EEC, is seeking to establish a quicker and lightweight authentication mechanism in IoT, the second method, is an explicit certificateless implement using rivest-shamir-adleman algorithm (RSA), the latter is slower than the first method, because the ECC algorithm is better in storage requirements and processing resources, given the length of the key used, and provides an equivalent security degree. On another side, Shamir's secret sharing method (SSS) has been applied as the mutual authentication mechanism in [13], to authenticate the communication among the nodes in wireless sensor network (WSN), the mechanism involved building a secret key and mapping it, using the SSS method, on the working nodes, to become a portal to achieve the mutual authentication in the network.

Despite the several benefits for ECC, this usage doesn't realize a strong enough mechanism unless resolving the standard ECDSA's problem, i.e., the reuse selection randomness key $(k)$, this weakness makes it vulnerable to exposing the private key. Several researchers tried to manipulate this weakness. In [14], Thomas pronin suggests deterministic methods to calculate $k$ for each transaction rather than choice randomly, the deterministic scheme, in a specific aspect, depends on the hash value of the message, so this solution considered specific applications, such as Bitcoin, and there are no guarantees to apply it in different applications such as applications that involves a lot of messages sent as control signals, for example, when user object request some critical information from temperature sensor, by sending constant queries message as a remote-control . SHUANG-GEN LIU [15] proposed generating double signature's private keys in the

signature creation phase, one of them used to generate a digital signature, and another used in the verification procedure, also, this proposed eliminating the modular inverse calculation in signature and verification phases to decreases the algorithm's complexity and enhance its run time. The theoretical analysis proves that the random integer decomposition into two integers couldn't throw out the random integer reusing weakness, since this

Scheme leads to retrieving the private key when used twice. Nisreen T. Hussein [16] proposed using a double elliptic curve's (EC) private key, that yields to generate double public keys for each user. In another word, by modifying the standard ECC's encryption parameters by selecting double private keys, and calculating double public keys, this method resolved the reused of random selection integers in ECDSA, but doubly public keys by performing additional point multiplication operations could be made the execution time worse. Also increasing the key length require more memory capacity. And these results don't accept in limited-resources environments.

The Contribution. Taking into consideration all the above ECDSA's weakness, this paper presents a modified ECDSA, alongside the SSS method to build a hybrid algorithm, the proposed algorithm contributions can be summarized as follow:

1. addition to resolving the randomness vulnerable of the ECDSA, the modification can be given both a more reliable solution and high-performance.
2. as well, this algorithm has a perfect dealing with the system scalability, this is an important concern, as long as, the IoT networks may be rapidly expanded to meet the increasing users' requirements.
3. The hybrid algorithm provides distributed computations to decrease the efforts on the individual device's resources, and make the authentication model more robust, lightweight and suitable to apply in limited-resources environments.

The remainder paper's organization is: section 3 presents the general construction of ECDSA, and Shamir's secret sharing algorithm, including a theoretic analysis of ECDSA's weakness. Section 4 presents a general view of the IoT-physical network design and a depth view of the security levels of the proposed algorithm with their distributions and calculations. The security analysis and the results are discussed in section 5. Section 6 involved performance and security comparisons with other related works. Finally, section 7 shows the conclusions.

Table 1. Notations

| Notation | Description |
|---|---|
| $p$ | Prime number |
| $GF_p$ | Prime field |
| $E_p$ | The elliptic curve defined over a finite field $GF_p$. |
| $(a, b)$ | EC's parameters $\in GF_p$ |
| $G$ | A base point on the curve $E_p(a, b)$ |
| $Q_A$ | A public key is a point residing on the curve $E_p(a, b)$ |
| $n$ | Order of the $G$, i.e., $n \times G = o$ |
| $h$ | Cofactor defined as $h = \frac{E_p}{n}$ |
| $M$ | Message |
| $h(\ )$ | Secure hash function |
| $f(x_i)$ | Secret sharing function |
| $S$ | A secret value |
| $s_i$ | Pieces of the secret. |
| $NS$ | Sites that share a secret. |
| $T$ | Threshold value |
| $K$ | A random integer serves as a secret sharing parameter. |
| $k^{-1}$ | Inverse of $K$ |
| $k_i$ | Piece of the $K$. |
| $r$ | Global Authenticity Factor |
| $r^{-1}$ | Inverse of $r$ |
| $s_A$ | User digital signature serves as a Local Authenticity Factor |
| $s^{-1}$ | Inverse of $s_A$ |
| $query_{user}$ | The user's message involved a specific query. |
| $mod$ | Modular arithmetic |

## 3. Preliminaries

In this section, we presented the standard algorithm of ECDSA and some comparisons with other digital signature algorithms. Also, we will briefly point out the main concept of Shamir's secret sharing cryptosystem (SSSC) later.

### 3.1 Elliptic curve

Consider $GF_p$ as a prime field, where $p > 3$, and let (a, b) is two constants $\in GF_p$, then the Elliptic Curve $E_p(a, b)$ is a set of points, satisfied by the following non-singular equation:

$$y^2 = x^3 + ax + b \ (mod\ p) \ldots \quad (1)$$

$$where: 4a^3 + 27b^2 \neq 0 \ mod\ p \ldots \quad (2)$$

#### 3.1.1. Elliptic curve discrete logarithm problem[17]

The strength of the elliptic curve cryptography

Table 2. Comparison between ECDSA and other asymmetric digital signature algorithms

| Cryptosystems | Mathematical Problem | Length of key |
|---|---|---|
| RSA, Rabin-Williams | Integer Factorization | $> 1024$ bits |
| El Gamal, DSA, Diffie-Hellman | Discrete Logarithm | $== 1024$ bits |
| ECDSA, EC-Diffie-Hellman | ECDLP | $\leq 512$ bits |

ECC is how to find private random integer $d_A$, wherein $d_A G = Q_A$, that is easy to calculate $Q_A$ when known $d_A, and\ G$, but, when known $Q_A, and\ G$, is very impossible to deduce $d_A$. So, the ECC security core is:

$$d_A G = Q_A \ldots \quad (3)$$

As the result, the ECC security is dependent on the hardness of the discrete logarithm problem (DLP), and efficient methods to solve ECDLP have exponential complexity time[18].

Table 2 explain briefly some of ECDSA's advantages over other asymmetric digital signature algorithms[19].

#### 3.1.2. Elliptic curve digital signature algorithm (ECDSA)

Following the standard algorithm steps[20]:

1. **Initialization:** initialize the Elliptic Curve parameters: $(p, a, b, Q_A, n, d_A, h)$
2. **Sing generation:** at the sender side the signer does the following step to create a digital signature:

- Select random number $k \in [1, n-1]$.
- Calculate $(x, y) = kG \ldots (4)$.
- Calculate $r = x\ mod\ n \ldots (5)$.
- If $r == 0$, then reselect another $k$.
- Calculate $e = h(M) \ldots (6)$.
- Compute $s = k^{-1}(e + rd_A) mod\ n \ldots (7)$
- If $s == 0$, then reselect another $k$.
- The signature is the pair: $(\boldsymbol{r}, \boldsymbol{s})$.

### 3. Signature verification:

at the receiver, the following steps have to be taken to verify the previous sender's signature:

- All public parameters should store on the receiver side to be able to perform the verification process.
- Verify that $r\ and\ s$ are integers $\in [1, n-1]$, else, the signature is invalid.
- Calculate $e$ according to Eq. (6)… (both sides have the same secret $h()$).
- Compute:

$$u_1 = es^{-1}\ mod\ n \ldots \quad (8)$$

$$u_2 = rs^{-1}\ mod\ n \ldots \quad (9)$$

- Compute:

$$(x, y)' = u_1 G + u_2 Q_A \ldots \quad (10)$$

$$v = x' mod\ n \ldots \quad (11)$$

- The signature is valid if, and only if ($v = = r$), otherwise rejected.

### 4. Correctness of ecdsa:

Previously verification function is correct, according to Eq. (10).

$$(x, y)' = u_1 G + u_2 Q_A$$

$$(x, y)' = u_1 G + u_2 d_A G$$

$$(x, y)' = (u_1 + u_2 d_A) G$$

$$(x, y)' = (es^{-1} + rs^{-1} d_A) G$$

$$(x, y)' = (e + rd_A) s^{-1} G$$

$$(x, y)' = (e + rd_A)(k^{-1})^{-1}(e + rd_A)^{-1} G$$

$finally$: $(x, y)' = kG$ ... this equation of $(r)$

#### 3.1.3. Ecdsa weakness:

There are circumstances in that ECDSA can be vulnerable to revealing the private key $d_A$:

First circumstance: when reusing the random selection number $k$ with a different public key $Q_A$ [21]:

In which $r_1 = r_2 = r$, and $d_1 \neq d_2$, that means if an adversary stole the $k$ at a given session, then he/she can retrieve the private key of all subsequent sessions, according to Eq. (7).

$$s_1 = k^{-1}(e_1 + rd_1) mod\ n$$

$$s_2 = k^{-1}(e_2 + rd_2) mod\ n$$

So, we can calculate $d_i\ from$:

$$d_i = r^{-1}(k\, s_i - e_i) mod\ n \qquad (12)$$

Second circumstance: when reusing the random selection number $k$ with the same private key $d_A$, in which $d_1 = d_2 = d_A$, then, can calculate $k$ as follow:

$$k = (s_1 - s_2)^{-1}(e_1 + e_2) mod\ n \qquad (13)$$

### 3.2 Shamir's secret sharing cryptosystem (SSS):[22]

Divide specific data $D$ into multiple sub-data $d_i$ and distribute them onto several sides, making the original data $D$ un-reconstructable unless there are $T$ pieces that can reveal to find again the original data $D$, so any $T-1$ pieces cannot find it. The big advantage of SSS is adaptable to be applied inside distributed systems.

Such a concept depends on determining the threshold $T$ i.e., a value that defines the number of pieces of $d_i$ that is required to reconstruct the $D$ after dividing it and distributing it over multi sides. SSC algorithm consists of two phases:

#### 3.2.1. Distribution phase:

- Take some secret data denoted by $S$.
- Define the number of Sites $NS$ that receive the secret pieces $s_i$.
- Define the threshold value $T$, pieces of data from them we can reconstruct the $S$.
- Build the polynomial function $f(x_i), to\ calculate\ NS\ secret\ pieces,\ f(x_i)$'s degree is $T-1$, the constant part of $f(x_i)$ is the original Secret, and the $T-1$ coefficients of it are random integers chosen from $GF(S)$:

$$f(x_i) = \left(\sum_{j=0}^{T-1} (a_j \times x^j)\right) mod(S) \ldots \qquad (14)$$
$$\text{for } i = \{1, \ldots NS\}$$

$$where:\ \ a_0 = S\ , \{a_1, \ldots, a_{T-1}\} \in GF(S)$$

- After generating $NS$ pieces $= \{s_1, s_2, \ldots., s_{ns}\}$, distribute them onto Sites.

#### 3.2.2. Reconstruction phase:

Found out the $T$ pieces of data $s_i$ that require building the $S$.

Reconstruct the original $f(x)$ by using the LaGrange interpolation formula:

$$l_i = \prod_{M \neq j} \left(\frac{x - x_M}{x_j - x_M}\right) \qquad (15)$$

$$f(x) = \sum_{i=0}^{k-1} (y_i \times l_i) \qquad (16)$$

### 3.3 SSC with ECDSA, in IoT environments:

In IoT environments there are multi-object, these objects can be either user devices (requesting devices), or IoT devices (gathering and supplier information devices) that are connected directly with mini-computer devices (such as Raspberry pi), and surely there are centralized-server take the registration responsible, i.e., play the third trusted party role to make the communication between other objects securely and direct as possible as.

This scheme is a hybrid system combined with a centralized and distributed system, so, can leverage the features of SSS to support the ECDSA computations and we attempt to reduce the probability of exposing the private key of ECDSA, that was previously mentioned.

## 4. Proposed system design:

In the proposed system design, multiple objects will take specific roles for integration into the authentication model. The hybrid authentication infrastructure consists of two methods: ECDSA, and SSC, the stages are distributed over the system's objects to decrease the complexity of computations and communication overhead and provide a high degree of security against any vulnerable design in standard algorithms.

### 4.1 Physical design:

The system may be consist of multiple user devices such as PCs, laptops, smartphones, etc., that are used by users to request relevant information, such as photos, videos, sensor information, and else, from several IoT devices such as sensors, surveillance or security cameras (CCTV), physical access control systems, and so on, all these devices should have valid identities and registered at a centralized server, so, all interactions between system's objects should be authorized and in-control of uniform and comprehensive security model.

The experimental platform, used in this work involved a user laptop with a windows11, corei7 processor and 8GB RAM, the laptop was used as a user requester device and centralized server device at the same time, the camera module as an IoT device

connected directly with raspberry pi 3b+ containing ARM Cortex-A53 1.4GHz and 1 GB RAM.

## 4.2 The proposed algorithm

The main stages are:

#### 4.2.1. Registration stage:

To add a new user's device or IoT device should register these devices into our centralized server. The server is maintaining a registration database (RDB) to manage the device's information in the system. This information involved:

- Device's ID.
- Device type (IoT's or user's, device).
- User name (When the type is "user's device").
- Date of registration.
- Biometric attributes (such as the user's picture, or fingerprint)
- Other necessary information.

Fig. 1 shows, when a new user sends the primary information to the Server, likewise, when plugging a new IoT device.

#### 4.2.2. Authentication certificate stage:

After then, the server evaluates and verifies the correctness of the information that had been received, before storing them in RDB.

If it is verified, the authentication certificate is established for the registered object, this setup procedure is done as follows:

(Let's assume $A$ is an object)

**a)** The server creates elliptic curve $EC$ 's main parameters[23]:

**1.** Select large prime number $P$.

**2.** Initialize two random integers $a, b \in [1, P-1]$, as a $E_P$ coefficients.

**3.** Select base point $G$ that relies on the $E_P$, of order $n$.

**4.** Sends the $(E_P(a,b), G)$ parameters to $A$.

**b)** Authentication certificates creation:

**1.** Select secret random integer: $K \in [1, n-1]$, as the Secret Sharing parameter.

**2.** Compute $r$ global authenticity factor according to Eq. (5).

**3.** Build the first-degree secret polynomial according to Eq. (14) ( Since the threshold $T$ is equal to 2 always), and make the $a_0 = K$, and the coefficients $a_i \in [1, K-1]$ are random private integers.

For each $NS$ object compute: $k_i = f(x_i)$, $i \in [1, NS]$

**4.** Distribute the authenticity pair $(\boldsymbol{r}, \boldsymbol{k_i}, \boldsymbol{G})$ among $NS$ objects in the system.

If a new object, would be added to the system, step (b) is repeated to calculate a new authenticity pair according. Also, we can, optionally, repeat step (b) to enable the perfect forward secrecy (PFS) protocol at each predefined period-time, hence establishing a new session encryption key after each new session this scheme strengthens the model and makes the secret parameters unpredictable.

#### 4.2.3. Authentication communication creation stage:

On the user $A$ side when requesting information from one of the IoT devices, $A$ should do the following calculations:

**1.** Selects the private session key $d_A$ , hence, the public key $Q_A \; according \; to$ Eq. (3).

**2.** compute $e$ according to Eq. (6).

**3.** and then compute $s$ Local Authenticity Factor as:

$$s_A = e + rd_A \; mod \; n \tag{17}$$

So, the pair $(\boldsymbol{k_A}, \boldsymbol{r}, \boldsymbol{s_A})$ is that the authentication part of the $A's$ request, it will send to the intended IoT device.

#### 4.2.4. Authentication verification stage:

At the intended IoT devices the following verification procedures will be done to fulfil the $A'$s query:

**1.** Using Eq. (15) $and$ Eq. (16), reconstructing the original secret $K$ (since a given device has two secret shares $k_A \; and \; k_{IoT}$).

**2.** Computes $e$ according to the Eq. (6).

**3.** Calculates:

$$u_1 = Kes_A^{-1} (mod \; n) \tag{18},$$

$$u_2 = Krs_A^{-1} (mod \; n) \tag{19}$$

**4.** Finally:

$$(x, y)' = u_1 G + u_2 Q_A \; (mod \; n) \tag{20}$$

Verify that $r == x' mod \; n$.. if it is, the request will fulfil, otherwise will block the requested $IP$ as an adversary object, and send back an error.

Fig. 2 shows the procedures of creating authenticated user's query to request required information from a specified IoT device, and the verification steps that involved reconstructing the secret share $K$, the verification stage of the hybrid

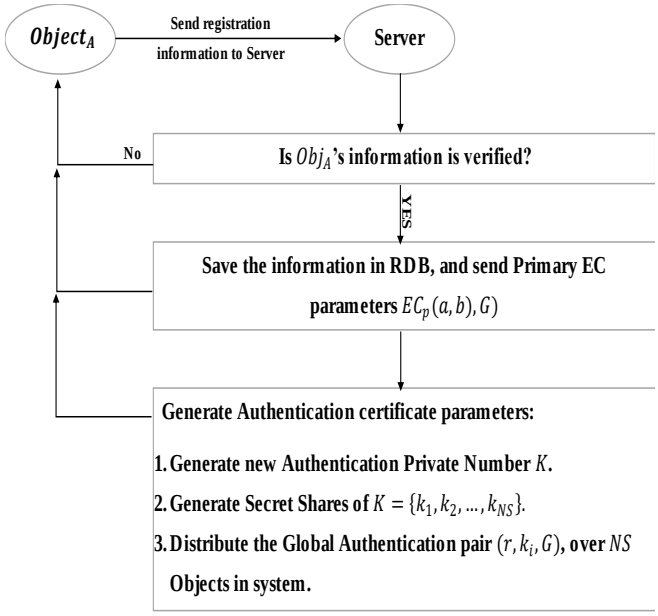


Figure. 1 The hybrid algorithm: registration and generate global authentication

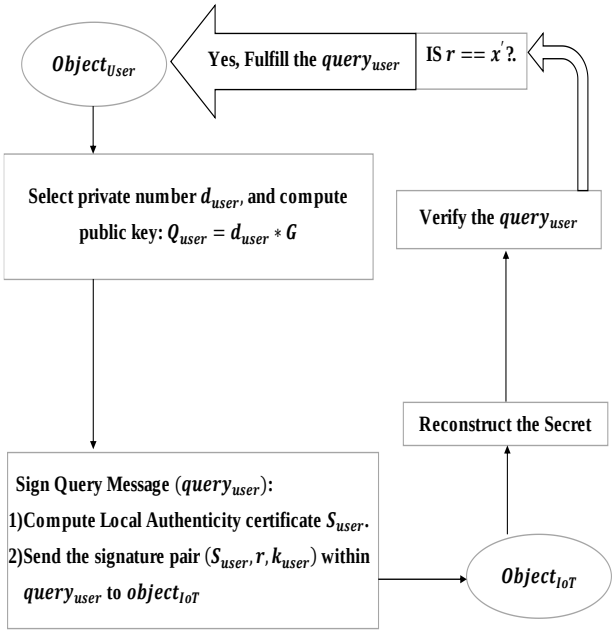


Figure. 2 The proposed algorithm: authenticated query, verification, and replay procedures

algorithm has variant computations that reduce the complexity of the modular inverse arithmetic to enhance the hybrid algorithm performance and make it more suitable to run in limited-resources environments.

**4.2.5. Correctness of the algorithm's verification calculations:**

The Eq.(20) is correct since:

According to Eq.(18) *and* Eq. (19):

$$(x,y)' = Kes_A^{-1}{}_1 G + Krs_A^{-1} Q_A$$

Since, $Q_A = d_A G$:

So, $(x,y)' = Kes_A^{-1}{}_1 G + Krs_A^{-1} d_A G$

Then: $(x,y)' = (e + rd_A)s_A^{-1} KG$

Since: $s_A = (e + rd_A)$

$$(x,y)' = (e + rd_A)(e + rd_A)_A^{-1} KG$$

Finally: $(x,y)' = KG$

And: $r == x - coord\ of\ (KG) mod\ n$

## 5. Security and efficiency analysis

The security features of the proposed algorithm demonstrate the strengthened aspects, that can be summarized in the following:

### 5.1 The ability to reuse the random selection integer$K$ (RRI):

Multiple times without exposing the private key because it would not be part of $\boldsymbol{s}$ in Eq. (17).

Let's assume that $A$ used the same parameters to create double queries:

Computes hash values of query messages, by Eq. (6):

$$e_1 = h(query_1)$$

$$e_2 = h(query_2)$$

Then:

$$s_A^1 = e_1 + rd_A\ mod\ n$$

$$s_A^2 = e_2 + rd_A\ mod\ n$$

$$by\ subtracting\ both\ s_A^1\ from\ s_A^2, will\ get:$$

$$s_A^1 - s_A^2 = e_1 - e_2$$

hence, all private parameters in the algorithm have been incomputable.

### 5.2 The scalability

The performance of the proposed algorithm doesn't affect by the system scalability, in another word, the devotion of the centralized server for a registration responsible, only, makes the installation of a new device in the network easier, which, the installation procedure involves sending the require

Table 3. Comparison between the complexity of the proposed algorithm and other algorithms

| Method | PG | SG | | SV | |
|---|---|---|---|---|---|
| | SM | SM | IM | SM | IM |
| [15] | 1 | 1 | | 2 | |
| [16] | 2 | 1 | 1 | 2 | 1 |
| The propose | 1 | 1 | | 2 | 1 |

Table 4. Comparison between the security of the the proposed algorithm and other algorithms

| Method | FA | RRI | Data tempering | Scalability | FSP |
|---|---|---|---|---|---|
| [15] | ✓ | ✗ | ✓ | ✗ | ✓ |
| [16] | ✓ | ✓ | ✓ | ✗ | ✓ |
| proposed | ✓ | ✓ | ✓ | ✓ | ✓ |

information to the server and accept the global authentication parameters to become capable to interact with other devices in the network.

### 5.3 forgery attack (FT)

If User C attempt to forge a signature, by calculating $s'$ from $r'as$:

$$s' = e + r'd_A$$

And send the forged signature $(r', k_C, s')$, the IoT device will reject the C's message, since:

$$Kes'^{-1}G + Kr_{IoT}s'^{-1}d_AG \neq (x, y)$$ (Invalid)

### 5.4 Forward secrecy protocol (FSP) and ECDLP complexity:

Even the attacker knows both $Q_A \; and \; G$, he can't deduce the private key $d_A$, because of the hardness of ECDLP. Consequently, this feature achieves the forward secrecy protocol, even if the attacker will get the private key of one of the legal parties, he can't get the session key.

### 5.5 Immunity against data tampering:

Using a hash function plays the important role in signature generation, so any data tampering can cause completely different results on the remote side. Hence, the sharing of a secret hash function h() among all legal parties allows for the reinforcement of the authentication model.

## 6. Performance and security comparisons:

As mentioned in sec.2, there are three research works [11, 15, 16], that proposed a modification to the standard ECSDA, these modifications are made to improve the run time efficiency, or to reduce the probability of the private key steal. The theoretical analysis for these modifications has been proven weakness points also.

Following a comparison between the proposed algorithm and the related works in [15, 16], the comparison takes into it account the number of cryptography computations as a benchmark, some operations don't take into account, because of the trivial cost (such as XOR operation). So, two operations that are considered in Table3, inverse multiplication of modular arithmetic (IM), and scalar (point) multiplication (SM), those taken place in three different stages, parameters generation stage (PG), sign generation stage (SG), and sign verification stage (SV). as shown, although the number of arithmetic operations that used in [13] is less by one, the proposed algorithm's operations are distributed over three sides, server, User, IoT sides, therefore, the overall complexity of these operations doesn't have the same overload that taken by [13].

Table 4 shows the proposed algorithm could ensure further important security features, that discussed in section 5. In contrast, the solution for the reused selected random integers problem is missed in [13]. On another hand, despite the [14] overcoming the reusing of random integers, this solution was coming with the expense of the algorithm complexity. Hence, adding an extra security level in the proposed model, through the SSS security level, can serve as a protection for the selected random integer by resolving it across multiple secret shares parts. In addition, these secret shares parts give another advantage, by which, the system can be efficiently deal with the scalable system.

## 7. Conclusion

In this paper, we have presented a proposed authentication model that focused on multiple aspects, the most important of which is how to get rid of a reusing random integer weakness in the original ECDSA, and at the same time, how to get better performance and lightweight model, so, as proven, the modification of ECDSA's calculations in signature generation and verification, that presented in the proposed algorithm can reduce the probability of revealing a secret key when reselected random integers, even if the same integer was used for signing different messages, in the same context, the proposed model takes advantage of Shamir's secret sharing

method to encrypt and distribute the secret (the selected random integer) and in addition to the encrypt purpose, The distribution of the secret will reduce the calculations efforts on multiple sides, which will make the algorithm quickest and less complexity. As result, the proposed algorithm takes into consideration the limitations in storage, and processing power, in the IoT environments, when resolving the weakness in ECDSA and presenting a new version of the algorithm.

## Conflicts of interest

The authors declare that there is no conflict of interest.

## Author contributions

Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing-original draft preparation, writing-review and editing, and visualization have been implemented by the first author. Supervision, and project administration, have been implemented by the second author.

## References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey", *Comput. Networks*, Vol. 54, No. 15, pp. 2787–2805, 2010.

[2] Z. Yang, J. He, Y. Tian, and J. Zhou, "Faster Authenticated Key Agreement with Perfect Forward Secrecy for Industrial Internet-of-Things", *IEEE Trans. Ind. Informatics*, Vol. 16, No. 10, pp. 6584–6596, 2020.

[3] J. Mo and H. Chen, "A Lightweight Secure User Authentication and Key Agreement Protocol for Wireless Sensor Networks", *Secur. Commun. Networks*, Vol. 2019, 2019.

[4] M. Taufiq and D. Ogi, "Implementing One-Time Password Mutual Authentication Scheme on Sharing Renewed Finite Random Sub-Passwords Using Raspberry Pi as a Room Access Control to Prevent Replay Attack", In: *Proc. of 2nd 2018 Int. Conf. Electr. Eng. Informatics, ICELTICs 2018*, pp. 13–18, 2018.

[5] Minahil, M. F. Ayub, K. Mahmood, S. Kumari, and A. K. Sangaiah, "Lightweight authentication protocol for e-health clouds in IoT-based applications through 5G technology", *Digit. Commun. Networks*, Vol. 7, No. 2, pp. 235–244, 2021.

[6] T. Yang, G. H. Zhang, L. Liu, and Y. Q. Zhang, "A survey on authentication protocols for internet of things", *J. Cryptologic Res.*, Vol. 7, No. 1, pp. 87–101, 2020.

[7] S. Rajesh, V. Paul, V. G. Menon, and M. R. Khosravi, "A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices", *Symmetry (Basel).*, Vol. 11, No. 2, 2019.

[8] S. A. Fadhil, "Internet of Things security threats and key technologies", *J. Discret. Math. Sci. Cryptogr.*, Vol. 2021, 2021.

[9] M. S. Fadhil, A. K. Farhan, and M. N. Fadhil, "A lightweight aes algorithm implementation for secure iot environment", *Iraqi J. Sci.*, Vol. 62, No. 8, pp. 2759–2770, 2021.

[10] A. Zahan, M. S. Hossain, Z. Rahman, and S. K. A. Shezan, "Smart home iot use case with elliptic curve based digital signature: An evaluation on security and performance analysis", *Int. J. Adv. Technol. Eng. Explor.*, Vol. 7, No. 62, pp. 11–19, 2020.

[11] Y. Genc and E. Afacan, "Design and implementation of an efficient elliptic curve digital signature algorithm (ECDSA)", In: *Proc. of 2021 IEEE Int. IOT, Electron. Mechatronics Conf. IEMTRONICS 2021*, 2021.

[12] D. H. Lee and I. Y. Lee, "A lightweight authentication and key agreement schemes for IoT environments", *Sensors (Switzerland)*, Vol. 20, No. 18, pp. 1–18, 2020.

[13] S. VenkataRao and V. Ananth, "A Hybrid Optimization Algorithm and Shamir Secret Sharing Based Secure Data Transmission for IoT based WSN", *Int. J. Intell. Eng. Syst.*, Vol. 14, No. 6, pp. 498–506, 2021, doi: 10.22266/ijies2021.1231.44.

[14] P. Ipa and S. Di, "Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)", *Gospod. Mater. i Logistyka*, Vol. 26, No. 4, pp. 185–197, 2017.

[15] S. G. Liu, W. Q. Chen, and J. L. Liu, "An Efficient Double Parameter Elliptic Curve Digital Signature Algorithm for Blockchain", *IEEE Access*, Vol. 9, pp. 77058–77066, 2021.

[16] N. T. Hussein and A. H. Kashmar, "An Improvement of ECDSA Weak Randomness in Blockchain", *IOP Conf. Ser. Mater. Sci. Eng.*, Vol. 928, No. 3, 2020.

[17] K. A. Chavan, I. Gupta, and D. B. Kulkarni, "A Review on Solving ECDLP over Large Finite Field Using P arallel Pollard ' s Rho ( ρ ) Method", *IOSR J. Comput. Eng.*, Vol. 18, No. 2, pp. 1–11, 2016.

[18] T. M. Aung and N. N. Hla, "A Study of General Attacks on Elliptic Curve Discrete Logarithm

Problem Over Prime Field and Binary Field", *SSRN Electron. J.*, No. November, 2018.

[19] F. Mallouli, A. Hellal, N. S. Saeed, and F. A. Alzahrani, "A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms", In: *Proc. of 6th IEEE Int. Conf. Cyber Secur. Cloud Comput. CSCloud 2019 5th IEEE Int. Conf. Edge Comput. Scalable Cloud, EdgeCom 2019*, pp. 173–176, 2019.

[20] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)", *Int. J. Inf. Secur.*, Vol. 1, No. 1, pp. 36–63, 2001.

[21] Z. Wang, H. Yu, Z. Zhang, J. Piao, and J. Liu, "ECDSA weak randomness in Bitcoin", *Futur. Gener. Comput. Syst.*, Vol. 102, pp. 507–513, 2020.

[22] A. Shamir, "How to Share a Secret", *Commun. ACM*, Vol. 22, No. 11, pp. 612–613, 1979.

[23] Certicom Research, *Standards for efficient cryptography, SEC 1: Elliptic Curve Cryptography*, Stand. Effic. Cryptogr., Vol. 1, No. Sec 1, pp. 1–22, 2009.