# MEKDA: MULTI-LEVEL ECC BASED KEY DISTRIBUTION AND AUTHENTICATION IN INTERNET OF THINGS

Padmashree M G, Mallikarjun J P, Arunalatha J S and Venugopal K R

Department of Computer Science and Engineering, University Visvesvaraya
College of Engineering, Bangalore University, Bengaluru, India

## *ABSTRACT*

*The Internet of Things (IoT) is an extensive system of networks and connected devices with minimal human interaction and swift growth. The constraints of the System and limitations of Devices pose several challenges, including security; hence billions of devices must protect from attacks and compromises. The resource-constrained nature of IoT devices amplifies security challenges. Thus standard data communication and security measures are inefficient in the IoT environment. The ubiquity of IoT devices and their deployment in sensitive applications increase the vulnerability of any security breaches to risk lives. Hence, IoT-related security challenges are of great concern. Authentication is the solution to the vulnerability of a malicious device in the IoT environment. The proposed Multi-level Elliptic Curve Cryptography based Key Distribution and Authentication in IoT enhances the security by Multi-level Authentication when the devices enter or exit the Cluster in an IoT system. The decreased Computation Time and Energy Consumption by generating and distributing Keys using Elliptic Curve Cryptography extends the availability of the IoT devices. The Performance analysis shows the improvement over the Fast Authentication and Data Transfer method.*

## *KEYWORDS*

*Authentication, Elliptic Curve Cryptography, Internet of Things, Key Distribution, Security.*

## 1. INTRODUCTION

The Internet of Things (IoT) has mainstreamed portable Internet applications and lead to the standardization of Communication protocols [1]. IoT strategy is incorporated in modern applications where the System exhibits limitations on communication overhead and information transmissions [2], [3]. IoT device obtains access to the System through the System Network. Authentication is an important and initial activity in security measures and data communication functionalities. Many connected nodes that communicate with each other have limited computational and battery power. The existing Authenticating and Key Distribution protocols to validate the IoT devices and the systems lead to more resource utilization *viz.*, Communication cost, Memory, and Energy consumption further increases when multiple IoT devices initiate concurrently.

The conventional information exchange framework uses the Cryptographic Key to secure communicated information. When Data transfers through various nodes, the traditional mono Key methodology is inappropriate. If this Key is compromised, then the complete communication framework is compromised. The disclosure is enormous, as a high volume of information transmission occurs in an IoT framework. Thus, constant verification and security are

DOI: 10.5121/ijcnc.2021.13602

predominant, complicated, and time-consuming. Robust security protection with identity privacy preservation [34] and non-repudiation uses Bilinear Pairing.

IoT networks usually have triple-layer architecture, including the Physical Sensor, Network, and Application layer. IoT nodes with different sensor characteristics are in the Physical layer. The Network layer enables servers to receive the sensor data transmitted to them. In general, gateways, routers, and packet-transfer devices connect the Application and the Physical layer. Standard Authentication methods are efficient in gateways because these devices have high computing power. Physical Sensor Layer nodes, conversely, need lightweight authentication solutions.

The security of IoT depends mainly on the secrecy of the Group Key. In dynamic networks, the devices frequently change the Groups leading to difficulty in securing shared crypto keys within a group remarkably, a Symmetric-Key. Robust key management protocols require Forward and Backward Secrecy and ensure immunity to Collusion attacks [4]. The compromised Group Key results in IoT functionalities exposed to attacks *viz*., Man-In-The-Middle, Denial-of-Service, and Replay attack [5], [6], [7].

*Motivation*: Authentication and Data transmission framework secure and validate the entire IoT Group based on complete Signing and Encryption [1]. Key Generator Center generates partial private Keys for IoT devices and avoids the Key Escrow problem [8]. The path among Devices, Gateway and Server, is dependent on the network domain security. The inter Group secure data transmission is not considered. A Single Key generated by the Key Generator Center for Group Communication is vulnerable to IoT Group Key Compromise Attack. The proposed Multi-level ECC-based Key Distribution and Authentication in IoT enhances the security by authenticating the communicating resource-constrained devices using Dynamic Aggregate Cluster Key and integrating Authentication and Data transmission.

*Contributions:* The main contributions of the proposed Multi-level Elliptic Curve Cryptography (ECC) based Key Distribution and Authentication in IoT are:

i) Authenticate and enhance security using Multi-level Authentication when an IoT device enters or exits the Cluster.
ii) Decrease the Computation Time and Energy Consumption and extend the availability of the IoT devices by accomplishing Key distribution using Elliptic Curve Cryptography.

*Organization*: The paper is organized as follows: Section 2 presents the works Related to the Cluster Key Distribution and Authentication; Section 3 abstracts the Background Work. Section 4 describes the Proposed Multi-level ECC-based Key Distribution and Authentication in IoT. Section 5 analyses the Performance and security of the proposed MEKDA, and Section 6 concludes the work.

## 2. Related Works

The Authentication protocol using simple functions of Cryptography *viz*., the Advanced Encryption Standard, and Hashing [9], the inter-network compatible security algorithm with scalable and portable IoT device provides a solution to the issues above. Authentication deals with the stolen devices without compromising the System services. The use of self-learning Crossbar Adaptive Array Artificial Intelligence technique protects from data disclosure. An Authentication and Data Sharing strategy [1] for an IoT environment allows multiple authentications while transmitting the data between the IoT devices and the network. Mutual Authentication is accomplished using the Aggregate Signing and Encryption process. The

Aggregated Sign Encryption method is resistant to Man-In-The-Middle and Replay attacks. The use of a single Group Key leads to a vulnerable communication system. The Lattice Encryption technology resists the Quantum Attack [9]. Mirai Attack, which controls the Actuators, can be overcome using Authentication and Encryption [10].

Lin *et al.*, [11] proposed a Twin layered Authentication approach using a lookup Attribute table. A Topology Control Mechanism is used to distribute the Device and the Gateway Keys; Authentication is performed using token or device information of the IoT device and the Gateway managed by the Server withstanding the Denial-of-Service Attack. The IoT nodes are identified by the Communicational attributes within the Group and by Physical Attributes for inter Group device authentication. The approach authenticates for the IoT devices having identical Attributes and provides security from the attacker while initializing the System.

Dammak *et al.*, [12] presented a Decentralized Keys Exchange framework to provide secure communication between the authenticated users of IoT environment using Master Token Encryption in a Hierarchical Framework. The Single Group Key and multiple Member Keys are updated when IoT devices change the Group association without the contribution of the Members. Forward/Backward Secrecy is achieved with resistance to the Collusion Attack. Data is accessible when the group member is compromised. Xu *et al.*, [13] constructed a Searchable Public Key Encryption method that generates the Ciphers comprising embedded structured Keywords using a single multiply and pair functionality. The Cloud recognizes the Ciphers by searching the Keyword List disclosed partial Keys.

Mansour *et al.*, [4] introduced a Centralized Dynamic Group Symmetric and Asymmetric Key Management approach using the Prime Number Factoring method. A single disconnected key is generated simultaneously during registration. The computation time and the size of the Group Key are directly proportional to the number of registered Devices in the Group. Aydin *et al.*, [6] propose a Centralized/Decentralized Group Authentication strategy with Symmetric Group Key Encryption. An Elliptic Curve Diffie-Hellman Group Key Exchange distributes the Group Keys among the Devices with Private Keys. Every Group Member shares the Partial Public Key with all the Group members. The Authentication leads to an Encrypted Private Key sharing process within the Group to obtain the Group Key. Thus security of the System depends on the Authentication process.

Cheng *et al.*, [14] designed a Group Member Authentication and Paired Key Establishment strategy using Polynomial based Tokens. The scheme is resistive against Internal and External Attacks. The use of XOR encryption and the Horner Polynomial Evaluation method reduces the computation time. Zhang *et al.*, [15] propounded an Anonymous Batch Authentication protocol using Certificates generated using Simplex Hash Chain. The Device-wise Certificates are maintained in a table for distribution while protecting the active Certificates. The authenticated IoT devices are identified, and Dual Hash Seeds are generated for unauthenticated devices. Mutual Authentication is achieved with resistance to Replay and Man-In-The-Middle Attack.

Wu *et al.*, [16] proposed a Group based Encryption/Decryption Scheme for nested Groups with Quad-Decryption using Single Key. The Group Key is generated and divided into Partial Keys to distribute among the Group Members. The Partial Group Keys are shared among the Group members using Second-degree Polynomial and recovered using Lagrange Interpolation. The threshold value restricts the involvement of all the Group Members in Group Key generation.

Qiu *et al.*, [17] proposed a Sign Encryption method to multicast the multiple data using Elliptic Curve Cryptographic Scalar Point Multiplication functionality. This method secures data transfer involving the Device and the Key Generating Authority. The Certificate-less Receiving Node

Identity is verified by the Gateway to reduce the computation cost. The Authentication framework reduces the security of the data, as the Authenticity of the Receiver depends on the Gateway. The authentication protocols use range [18], ECC [19], distributed [20], Unclonable Function [21], Polynomial [22], Mono-Input Multi-Output [23], Cross-Domain-oriented [24] Key management [25], [26]. Table 1 summarises the related works.

Table 1. Comparison of Related Works

| Author | Approach | Advantages | Disadvantages |
|---|---|---|---|
| Dammak *et al.,* [12] (2020) | Decentralized Keys Exchange, Hierarchical Master Token Encryption. | Single Group Key and multiple Member Keys updated IoT devices change the Group association; Forward/Backward Secrecy; resistance to the Collusion Attack. | Data is accessible when the group member is compromised |
| Mansour *et al.,* (2020) | Centralized Dynamic Group Symmetric and Asymmetric Key Management, Prime Number Factoring | A single disconnected key is generated simultaneously during registration. | The computation time and the size of the Group Key are directly proportional to the number of registered Devices in the Group. |
| Wu *et al.,* [4] (2019) | Group based Quad-Decryption using Single Key; Second-degree Polynomial, Lagrange Interpolation | Group Key into Partial Keys distribute to Group Members. | The Threshold value restricts the involvement in Group Key generation. |
| Qiu *et al.,* [16] (2019) | multicast multiple data using Elliptic Curve Cryptographic Scalar Point Multiplication | Secure data transmission between the IoT Device and the Key Generating Authority; Certificateless Receiving Node Identity is verified by the Gateway reduce the computation cost. | reduces the security of the data: Authenticity of the Receiver depends on the Gateway. |
| Cao *et al.,* [1] (2019) | Aggregate Signing and Encryption process. | resistant to Man-In-The-Middle and Replay attacks. | single Group Key may be vulnerable |

# 3. Background Work

## 3.1. Low-Energy Adaptive Clustering Hierarchy

Low-Energy Adaptive Clustering Hierarchy (LEACH) is a routing protocol where nodes transmit to Cluster Leader (CL), and the CL aggregate, compress and forward data to the Gateway. It assumes each node consists of sufficient transmission power to reach the Gateway or the nearest CL directly. But always using the transmission in its entirety dissipates energy. The repeated participation of CL, $q$, is the preferred quantum of nodes selected as CLs. $1/q$ is the possibility of each node that repeats as a CL. Every iteration ends with the non-CL nodes joining the Cluster with nearby CL. The nodes transmit data following the time slice that CL provides. The nodes use the least energy to reach the CL. The transmission power is switched on only in a given time slice.

*Characteristics:* LEACH selects CL randomly using a Threshold value $t_n$, or sensor with more energy. Given n non-Leader devices in $i^{th}$ iteration with $q$ quantum of participation for CL selection, the threshold $0 < t_n < 1$ is in equation (1).

$$t_n = q/(1 - q(\ i \ mod \ 1/q) \qquad (1)$$

Every non-Leader device not participated in the last $1/q$ iterations generates a random number $r \in [0, 1]$. If $r < t_n$, then device n is designated as CL in the current iteration. The non-CL devices attach to the nearest CL. Data aggregates at the CL. CL communicates directly with Gateway or the User. Devices communicate with CL is *via* Time Division Multiple Access. Each Cluster uses a different Code Division Multiple Access codes set to minimize intervention between clusters.
*Drawbacks:* CL selection ignores the residual energy of the devices. One-hop CL Gateway transmissions increase the energy usage leading to the death of CL disconnecting the Cluster. Time-based data transmission of LEACH protocol and heterogeneous and mobile characteristics of IoT devices makes it inappropriate for IoT scenarios.

### 3.2. K-Nearest Neighbor Clustering

K-nearest neighbors (KNN) is a simple, supervised clustering Machine Learning algorithm that compares attributes to infer values of new data points. *i.e.*, the point assignment is derived from its association with the other points. KNN clustering determines the distance from current data to each row of existing data using Euclidean, Manhattan, or Minkowski distance. The Data is distance-wise sort in ascending order and select the upper K rows. The new point categorizes into a recurrent group of $K$ rows. Given two points $(P_x, P_y)$ and $(Q_x, Q_y)$ at $d$ dimension, Euclidean Distance is the shortest distance between two points given by equation (2) for 2-dimension and equation (3) for $n$ dimensions.

$$dist_{Euclidean} = \surd\ (P_x - Q_x)^2 + (P_y - Q_y)^2 \qquad (2)$$
$$dist_{Euclidean} = \surd\sum{}^{n}{}_{d=1} (P_d - Q_d)^2 \qquad (3)$$

Manhattan Distance is the sum of absolute dimensional distance between two points given by the equation (4) and equation (5) for 2 and $n$-dimensions respectively.

$$dist_{Manhattan} = |\ P_x - Q_x\,| + |P_y - Q_y\,| \qquad (4)$$
$$dist_{Manhattan} = \sum{}^{n}{}_{d=1} |P_d - Q_d\ | \qquad (5)$$

Minkowski Distance is the universal representation of Euclidean and Manhattan Distance with norm order $N$ given by equation (6).

$$dist_{Minkowski} = {}^{N}\surd\sum{}^{n}{}_{d=1} \mid P_d - Q_d|^{N} \qquad (6)$$

It suits Clustering and regression, works efficiently on multi-cluster issues. Low $K$ is responsive to extremes, and a high $K$ is resilient to extremes as it mediates more supporters to predict. The storage and computation cost is more with inefficiency for vast unrelated and irregular data and reduced scalability.

### 3.3. Problem Statement

Given the set of IoT devices, design an efficient Key Distribution and Authentication framework for the Internet of Things applications to secure data of communicating resource-constrained devices, the objectives of the proposed work are:

i) To enhance security and authenticate IoT devices by Multi-level Authentication when an IoT device enters or exits the Cluster.

ii) To extend the availability of the IoT devices by decreasing the Computation Time and Energy Consumption by accomplishing secure Key distribution using Elliptic Curve Cryptography.

## 4. THE PROPOSED MULTI-LEVEL ECC BASED KEY DISTRIBUTION AND AUTHENTICATION IN IOT

The Sending and Receiving Cluster components effectively encrypt and decrypt the message [30] using Elliptic Curve Cryptography [29], [32].

### 4.1. System Architecture

The architecture of Multi-level ECC-based Key Distribution and Authentication in the IoT is shown in Figure 1. The nodes group into Clusters in the network. Each Cluster consists of a Cluster Leader. The Cluster Leader comprises more computation power compared to other Cluster nodes. The Cluster Leader is used for inter Cluster communication to make the overall network energy-efficient and efficiently use the energy of each node.

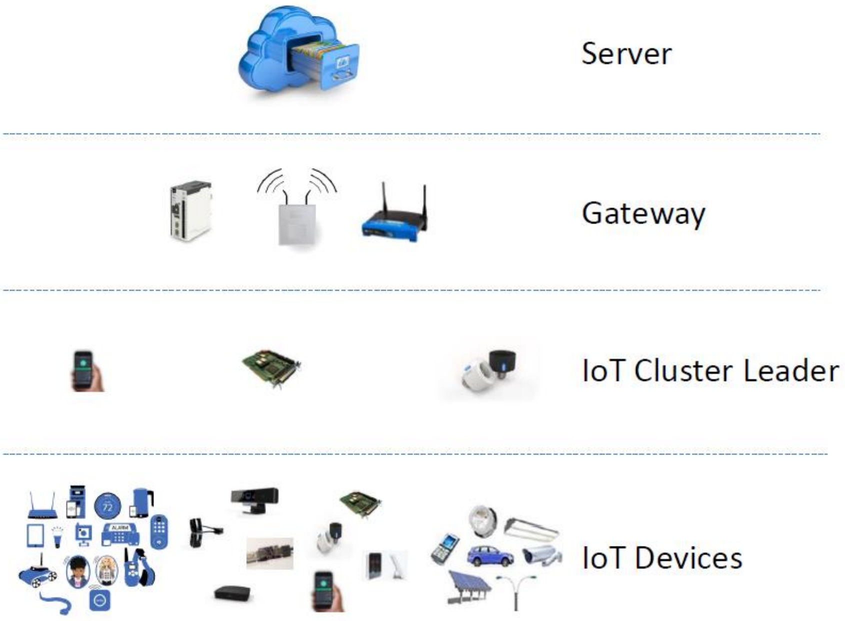

Figure 1. The Architecture of the Proposed MEKDA

### 4.2. System Block

The Block Diagram of the Multi-level ECC-based Key Distribution and Authentication (MEKDA) in IoT is given in Figure 2.

The Initialization, Cluster Formation, Elliptic Curve Coordinators selection is processed before the Cluster Head selection. The Coordinator is the Auxiliary Cluster Leader that overtakes the group on Cluster Head energy below the threshold. The Authentication and Data Transmission module is processed after the Cluster Key generation and Exchange between the communicating devices.

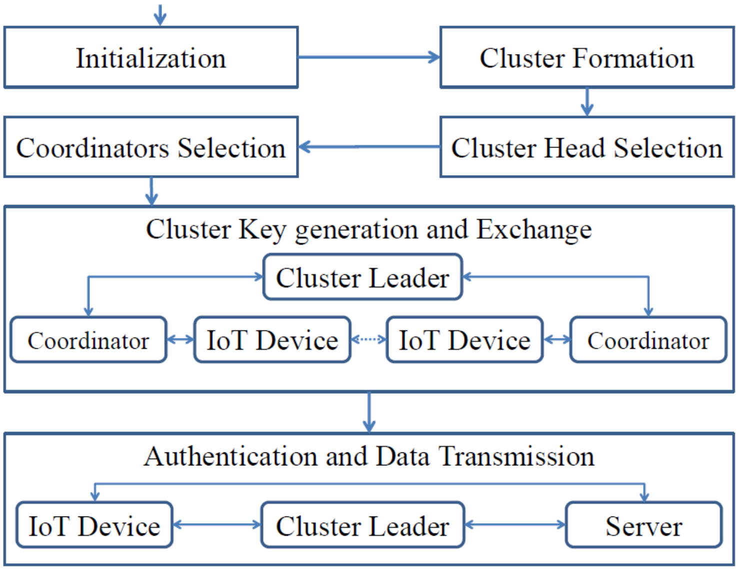

Figure 2. Block Diagram of the Proposed MEKDA

### 4.3. The proposed Multi-level ECC based Key Distribution and Authentication (MEKDA) Algorithm

MEKDA scheme consists of Setup, Signup, and Authentication modules. The Server (*S*) performs Setup, Sign in, and Signup modules. Device *D* interacts with Gateway *G* to secure mutual Authentication and session key distribution. Cluster Auxiliary Leader device communicates with *G via* an intermediary Cluster Leader device *CL*. Cluster Leader device *D* directly communicates with *G* by eliminating the intermediary device *CL*. The proposed MEKDA protocol is based on ECC and a Hash function $h : 0, 1^* \rightarrow 0, 1^l$. Table-2 defines the notations.

Table 2. Table of Notations

| **Symbol** | **Definition** |
|---|---|
| $X_k$ | Exclusive Primary Secret Key |
| $X_{id}$ | Exclusive Secret Identity |
| $X_{id'}$ | Diminutive Distinct Identity |
| $X_a$ | Identity with Hashed Secret |
| $X_b$ | Identity with Shared Secret |
| *CAL* | Cluster Auxiliary Leader device |
| *CL* | Cluster Leader devices |
| $X_t$ | Timeframe |
| $X_{tid}$ | Transient identity |
| $t^*$ | received Time of the Message |
| $\Delta t$ | Maximum Transmission Delay |
| $h$ | Hash |
| $s_{k*}$ | Session Key |
| $r$ | Random Number |
| $t_n$ | Threshold value |
| $D_r$, $D_k$ | Elliptic Curve Points |

*Setup module*: The Server (S) configures the Gateway (G) in Function 1. The *S* selects a primary secret key $G_k$ for G and stores $G_k$ in G. It configures the Elliptic Curve parameters *viz.*, Generator Point, Curve Coefficients *etc.*, [32].

```
Function 1: Setup(), Setup
Input: $D_{id}$
Output: store $G_k$
The S selects a primary secret key $G_k$ for G
Stores $G_k$ in G
return
```

*Sign in module*: The Server and the nodes share the ECC Encryption/Decryption Keys [31] during *Sign in* to Encrypt and Decrypt messages during communication. The Elliptic-Curve-based encryption/decryption keys are shared among the devices to provide confidentiality.

*Sign up module*: The Device interacts with S and exchanges the ECC and other system parameters for Server authentication. The Server archives a device (D) as in Function 2:

```
Function 2: Signup(), Signup
Input: D
Output: $D_{id}$, $D_{id'}$, $D_a$, $D_b$
S selects an exclusive secret identity $D_{id}$ for D
S selects $D_k$ for D
S computes $D_a = D_{id} \oplus h(G_k||D_k)$ and $D_b = G_k \oplus D_a \oplus D_k$
S selects an additional diminutive distinct identity $D_{id'}$ for the Cluster Leader CL
S stores $\{D_{id'}, D_{id}, D_a, D_b\}$ in Cluster Leader device CL
S stores $\{D_{id}, D_a, D_b\}$ in the Cluster Auxiliary Leader device CAL
S stores the diminutive distinct identity $D_{id'}$ in Cluster Leader devices CL in G
return
```

$D_k$ is exclusive to compute $D_a$ and $D_b$. The identity $D_{id}$, is the stable and secret Key for device D. Cluster Leader device CL uses diminutive identity $D_{id'}$ only when functions as an intermediary. The module uses 3 Xor and a Hash function to store $D_{id} \oplus h(G_k||D_k)$ and $G_k \oplus\{D_{id} \oplus h(G_k||D_k)\}\oplus D_k$ in CAL and CL.

*Authentication module*: Every node interacts within the Cluster *via* the Cluster Leader and the Cluster communication *via* the Gateway. The use of a single cluster leader for authentication along with the Auxiliary Leader that acts secondary node increases the lifetime of the existing cluster. When the resources of the Cluster Leader fall below that of the Auxiliary Leader, the Cluster Leader transfers the control to the Auxiliary Leader *via* the Gateway retaining the Cluster. The use of ECC and Hashing improves the security level with irreversibility. The Cluster Leader verifies the Auxiliary Leader and Gateway verifies the Cluster Leader. Thus the scheme provides integrity, confidentiality in Cluster data communication.

```
Function 3: Authentication_D_CL()
Input: $D_{id}$
Output: $D_{tid}$, $D_y$, $D_a$, $D_b$, $D_t$
D selects $D_r$
D generates a time-frame $D_t$
D computes $D_x = D_a \oplus D_{id}$
D computes $D_y = D_x \oplus D_r$
D computes the Transient identity $D_{tid} = h(D_{id} \oplus D_t||D_r)$
return
```

The device D authenticates with the Gateway G *via* an intermediary device CL as in equation (7) and Function 3. It uses 3 Xor and 2 hash functions. $h(G_k||D_k)$, $h(G_k||D_k) \oplus D_r$ and $h(D_{id} \oplus D_t||D_r)$ sent to CL with a freshness [33] entity $D_t$ that addresses the Replay attack.

$$D \xrightarrow{\{D_{tid}, D_y, D_a, D_b, D_t\}} CL \qquad (7)$$

Equation (8) shows intermediate device CL forwarding data from device D to G after attaching the identity $CL_{id'}$ since G identifies CL using $CL_{id'}$, not $CL_{id}$. $CL$ forwards $h(D_{id} \oplus D_t||D_r)$, $h(G_k||D_k) \oplus D_r$, $D_{id} \oplus h(G_k||D_k)$, $G_k \oplus \{D_{id} \oplus h(G_k||D_k)\} \oplus D_k$ with authentication entity to $G$.

$$CL \xrightarrow{\{D_{tid}, D_y, D_a, D_b, D_t, CL_{id'}\}} G \qquad (8)$$

The Gateway transfers the authentication details *viz.*, $\{h(G_k||D_k)\}\oplus D_f$, $h(D_x||D_r||D_f\,||\eta||\mu)$, $\{ D_r \oplus D_f\} \oplus \{D_{id} \oplus h(G_k||D^+_k )\}$, and $\{ D_r \oplus D_f\} \oplus \{ G_k \oplus D_{id} \oplus h(G_k||D^+_k) \oplus D^+_k\}$ to the Cluster Leader as shown in equation (10) and Function 4 that uses 4 hash and 11 Xor. The validness of time-frame ($D_t$) verify using equation (9) to mitigates the Replay attack.

$$t^* - D_t > \delta t \qquad (9)$$

where, $t^*$ = receive time of the message and $\delta t = max(transmission\_delay)$.

$$G \xrightarrow{\{\alpha, \beta, \eta, \mu, CL_{id'}\}} CL \qquad (10)$$

```
Function 4: Authentication_G_CL()
Input: ${D_{tid}, D_y, D_a, D_b, D_t, CL_{id'}}$
Output: $\alpha, \beta, \eta, \mu, CL_{id'}$
if $CL_{id'}$ unknown to G then
    Abort
else
if ($t^* - D_t > \delta t$) then
    Abort
else
    G computes $D_{k*} = G_k \oplus D_a \oplus D_b$, $D_{x*} = h(G_k||D_{k*})$,
               $D_{id*} = D_{x*} \oplus D_a$, $D_{r*} = D_{x*} \oplus D_y$
G computes $D_{tid*} = h(D_{id*} \oplus D_t||D_{r*})$
if $D_{tid}$ <> $D_{tid*}$ then
    Aborts
else
    G selects $D_f$
    G computes $\alpha = D_x \oplus D_f$ and $\gamma = D_{r*} \oplus D_f$
    G selects a new $D^+_k$
    G computes $D^+_a = D_{id} \oplus h(G_k||D^+_k)$
    G computes $D^+_b = G_k \oplus D^+_a \oplus D^+_k$
    G computes $\eta = \gamma \oplus D^+_a$ and $\mu = \gamma \oplus D^+_b$
    G computes $\beta = h(D_x||D_r||D_f||\eta||\mu)$
    G computes and stores the session key, $Session\_Key = h(D_{id}||D_r||D_f||D_x)$
end if
end if
end if
return
```

CL forwards $\{h(G_k||D_k)\}\oplus D_f$, $h(D_x||D_r||D_f\ ||\eta||\mu)$, $\{D_r\oplus D_f\}\oplus\{D_{id}\oplus h(G_k||D^{+}_{k})\}$, and $\{D_r\oplus D_f\}\oplus\{G_k\oplus D_{id}\oplus h(G_k||D^{+}_{k})\oplus D^{+}_{k}\}$ received from G to D, detaching identity $CL_{id'}$ used for authentication as shown in equation (11) and Function 5.

$$CL \xrightarrow{\{\alpha,\ \beta,\ \eta,\ \mu\}} D \tag{11}$$

```
Function 5: Authentication_CL_D()
Input: $D_{tid}$, $\alpha, \beta, \eta, \mu$
Output: $\alpha, \beta, \eta, \mu$
D computes $D_{f*} = D_x \oplus \alpha$
D computes $\beta* = h(D_x||D_r||D_{f*}||\ \eta\ ||\ \mu)$
if $\beta <> \beta*$ then
    Aborts
else
    D computes $\gamma = D_r \oplus D_{f*}$
    D computes $D^{+}_{a} = \gamma \oplus \eta$
    D computes $D^{+}_{b} = \gamma \oplus \mu$
    D computes and stores the session key $s_{k*}$(= Session_Key) = $h(D_{id}||D_r||D_f||D_x)$
    D updates $(D_a||D_b)$ with $(D^{+}_{a}\ ||D^{+}_{b})$
end if
return
```

```
Algorithm 1: MEKDA, Multi-level ECC-based Key Distribution and Authentication
Input: Set of Devices
Output: Cluster Key Distribution and Authentication of Cluster Members
Initialize the System using Multi-attributed Nearest Neighbour Clustering Algorithm
Selects the Cluster Leader CL and CAL using LEACH algorithm with maximum resources within the Cluster
repeat
    IoT devices connect to nearby Registered Cluster Server via Gateway using Setup()
    S archives a device D using Signup()
    D sends Transitory identity to CL using Authentication_D_CL()
    intermediate CL forward Data with $CL_{id'}$ using Authentication_CL_G()
    G sends authentication details to CL using Authentication_G_CL()
    CL forwards the data to D detaching $CL_{id'}$ using Authentication_CL_D()
      if IoT Device Exits the Cluster, then
            ReKeying Triggered by CL and Distributed among the member IoT Devices
      end if
until all Clusters obtain the Cluster Keys
The Cluster Leader Authenticates the IoT device via the Server
if valid. then
    Sending IoT device Encrypts the data using Session Key
end if
The Cluster Leader Authenticates the IoT device via the Server
if valid, then
    Receiving IoT Device with Cluster Key Session Key Decrypts the Data
end if
return
```

Algorithm 1 shows the process of Multi-level ECC-based Key Distribution and Authentication.

Few selected Cluster Leaders aggregate the data from nearby members and forward them to the Gateway. Dynamic Cluster Leader aggregation and an Adaptive Clustering method reduce the energy overhead. The Setup state selects the CLs. The Consequent state maintains the CL when data transmits between the nodes. During the Setup state, deployment of nodes, broadcast input message from the Gateway. The nodes generate an arbitrary Number $r$ bounded by 0 and 1 and compares with the threshold value $t_n$. The node which generated the Random Number, if $r < t_n$, then becomes the CL; During the Consequent state, broadcasting of advertisement message to all other nodes from the elected CLs. The nodes other than the CLs determine to which Cluster it belongs, based on the strength of a received signal. After the Cluster Formation, Leaf nodes then transmit the sensed information to their CL. Then CL transmits the collected data to the Gateway. When the designated transmission time is over, start the setup phase to elect new CL, Advertising message from CLs to other nodes, and Cluster formation. During the Consequent state, a new cluster formation with a new node as the CL. Data aggregation at the CL and then to the Gateway takes place, and the process repeats. The sequential flow of the messages and the computations are depicted in Figure 3.

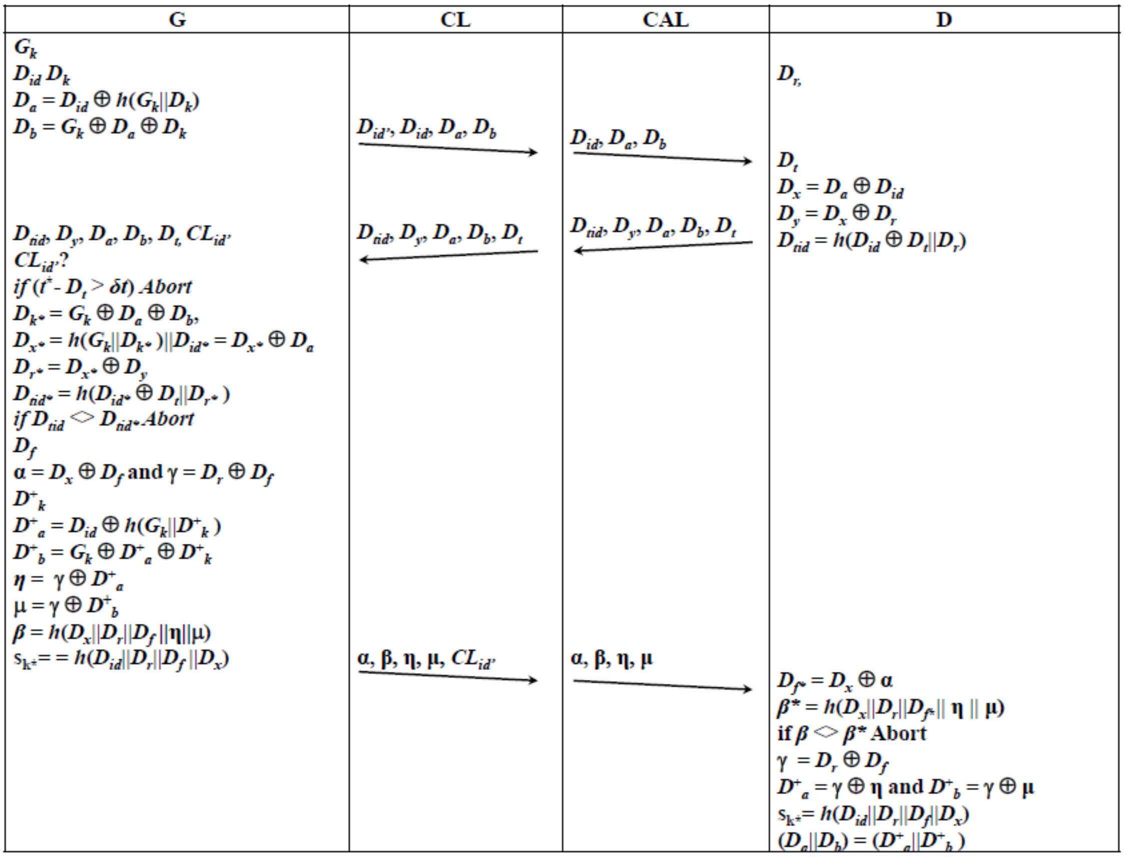
G
CL
CAL
D
$G_k$
$D_{id}\ D_k$
$D_a = D_{id} \oplus h(G_k \| D_k)$
$D_b = G_k \oplus D_a \oplus D_k$
$D_{r,}$
$D_{id'}, D_{id}, D_a, D_b$
$D_{id}, D_a, D_b$
$D_t$
$D_x = D_a \oplus D_{id}$
$D_y = D_x \oplus D_r$
$D_{tid} = h(D_{id} \oplus D_t \| D_r)$
$D_{tid}, D_y, D_a, D_b, D_t$
$D_{tid}, D_y, D_a, D_b, D_t$
$D_{tid}, D_y, D_a, D_b, D_{t,}\ CL_{id'}$
$CL_{id'}?$
$if\ (t^* - D_t > \delta t)\ Abort$
$D_{k^*} = G_k \oplus D_a \oplus D_b,$
$D_{x^*} = h(G_k \| D_{k^*}) \| D_{id^*} = D_{x^*} \oplus D_a$
$D_{r^*} = D_{x^*} \oplus D_y$
$D_{tid^*} = h(D_{id^*} \oplus D_t \| D_{r^*})$
$if\ D_{tid} <> D_{tid^*} Abort$
$D_f$
$\alpha = D_x \oplus D_f$ and $\gamma = D_r \oplus D_f$
$D^+_k$
$D^+_a = D_{id} \oplus h(G_k \| D^+_k)$
$D^+_b = G_k \oplus D^+_a \oplus D^+_k$
$\eta = \gamma \oplus D^+_a$
$\mu = \gamma \oplus D^+_b$
$\beta = h(D_x \| D_r \| D_f \| \eta \| \mu)$
$s_{k^*} = = h(D_{id} \| D_r \| D_f \| D_x)$
$\alpha, \beta, \eta, \mu, CL_{id'}$
$\alpha, \beta, \eta, \mu$
$D_{f^*} = D_x \oplus \alpha$
$\beta^* = h(D_x \| D_r \| D_{f^*} \| \eta \| \mu)$
if $\beta <> \beta^*$ Abort
$\gamma = D_r \oplus D_f$
$D^+_a = \gamma \oplus \eta$ and $D^+_b = \gamma \oplus \mu$
$s_{k^*} = h(D_{id} \| D_r \| D_f \| D_x)$
$(D_a \| D_b) = (D^+_a \| D^+_b)$

Figure 3. The sequential flow in the Proposed MEKDA

## 5. PERFORMANCE ANALYSIS

The proposed MEKDA for secure communication in IoT simplifies the authentication process and alleviates the load of the network, ensuring strong security protection, user anonymity, and non-repudiation. The simulation is conducted using java library SHA256Digest and EC256 in Java IDE on a 32-bit Ubuntu Platform with 1GB RAM. The 100 nodes are grouped into multiple Clusters. Every Cluster node interacts within the Cluster *via* the Cluster Leader and the Cluster communication *via* the Gateway using 256-bit Hashing Digest, 256 bit Elliptic Curve Java library as depicted in Figure 3. The parameters are stated in Table 3. The proposed MEKDA scheme withstands a variety of security attacks with ideal efficiency. The performance of MEKDA is compared with Fast Authentication and Data Transfer [1], Anonymous Mutual Authentication and Key Agreement Scheme (AMAKS) [27], Uniform Privacy Preservation Group Handover Authentication (UPPGHA) [28].

Table 3. Parameters

| | |
|---|---|
| Nodes | 100 |
| Area | 950x650m |
| Channel | Wireless |
| Transmission range | 150m |
| Initial Energy | 100Joules |
| Timestamp, id , tid | 64bits |
| Keylength | 256 bits |
| Hash | 256bits |

*Computation Time:* The Computation Time of MEKDA is given in Figure 4, Figure 5, Figure 6, and Table 4. The computation Time of MEKDA is 2 times of FADTS, 16 times of UPPGHA. Key distribution with a reduction in the number of ECC, Hash, and Xor decreases Computation Time. MEKDA uses 19 Xor, 8 Hash, and 2 scalar multiplications; FADTS uses 10 scalar multiplications and 6 hash; UPPGHA uses 10 exponential operations. MEKDA Computation Time is 4% more than AMAKS as it uses only 5 Hash and 17 Xor and ECC is irreversible but not Xor. Thus MEKDA authenticates and reduces computation time extending the availability of IoT devices providing confidentiality and integrity.

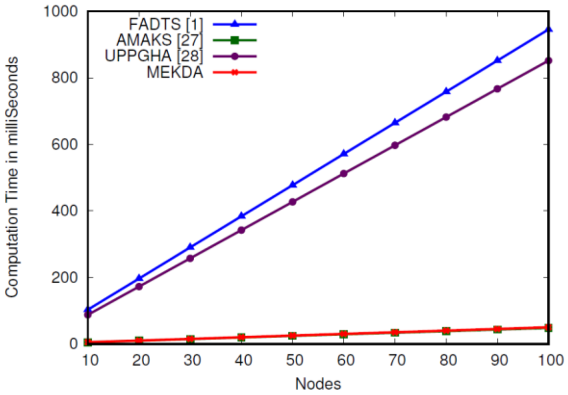

Figure 4. Computation Time of the Proposed MEKDA

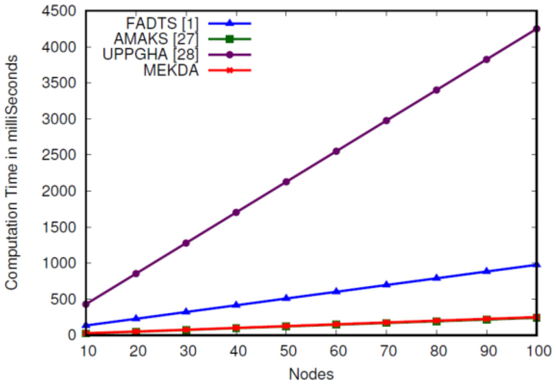

Figure 5. Computation Time of the Proposed MEKDA with group size 5

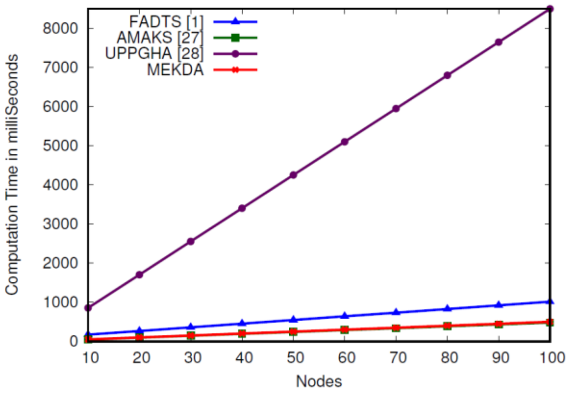

Figure 6. Computation Time of the Proposed MEKDA with group size 10

*Energy:* Figure 7, Figure 8, Figure 9, and Table 5 depict the Energy Consumed by MEKDA. The Energy Consumption of MEKDA is 2 times that of FADTS and 16 times that of UPPGHA. Key distribution with a reduction in the number of ECC, Hash, and Xor decreases energy consumption. MEKDA Energy Consumption is 4% more than AMAKS for 100 nodes of Group Size 1, 5, 10 as AMAKS uses limited 3 Hash and 6 Xor and ECC is irreversible but not Xor. The decrease in Energy Consumption extends the availability of the IoT devices by accomplishing Key distribution using Elliptic Curve Cryptography enhancing security by authenticating providing confidentiality and integrity.

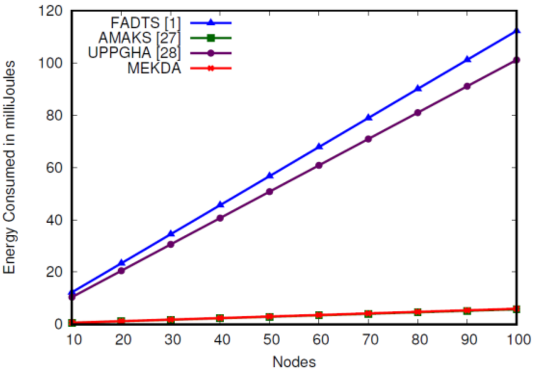

Figure 7. Energy consumed by the Proposed MEKDA

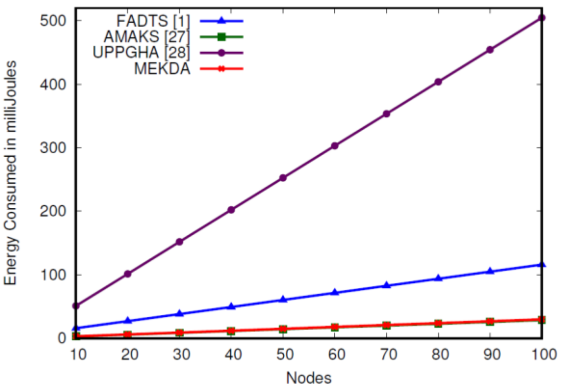

Figure 8. Energy consumed by the Proposed MEKDA with group size 5

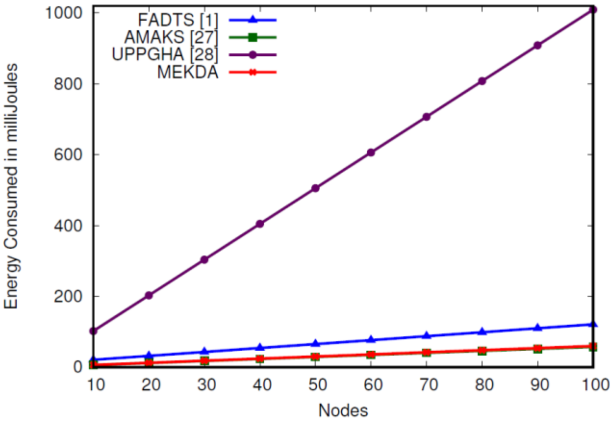

Figure 9. Energy consumed by the Proposed MEKDA with group size 10

Table 4. Computation Time (in milliSeconds)

| | FADTS | | | AMAKS | | | UPPGHA | | | MEKDA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Groups | 1 | 5 | 10 | 1 | 5 | 10 | 1 | 5 | 10 | 1 | 5 | 10 |
| 10 | 102.96 | 134.08 | 172.98 | 4.98 | 24.90 | 49.80 | 87.53 | 429.23 | 856.35 | 5.19 | 26.64 | 51.90 |
| 20 | 196.56 | 227.68 | 266.58 | 9.78 | 48.90 | 97.80 | 172.43 | 853.73 | 1705.35 | 10.19 | 51.06 | 101.93 |
| 30 | 290.16 | 321.28 | 360.18 | 14.58 | 72.90 | 145.80 | 257.33 | 1278.23 | 2554.35 | 15.20 | 76.12 | 151.96 |
| 40 | 383.76 | 414.88 | 453.78 | 19.38 | 96.90 | 193.80 | 342.23 | 1702.73 | 3403.35 | 20.20 | 101.18 | 201.98 |
| 50 | 477.36 | 508.48 | 547.38 | 24.18 | 120.90 | 241.80 | 427.13 | 2127.23 | 4252.35 | 25.20 | 126.24 | 252.01 |
| 60 | 570.96 | 602.08 | 640.98 | 28.98 | 144.90 | 289.80 | 512.03 | 2551.73 | 5101.35 | 30.20 | 151.30 | 302.04 |
| 70 | 664.56 | 695.68 | 734.58 | 33.78 | 168.90 | 337.80 | 596.93 | 2976.23 | 5950.35 | 35.21 | 176.36 | 352.06 |
| 80 | 758.16 | 789.28 | 828.18 | 38.58 | 192.90 | 385.80 | 681.83 | 3400.73 | 6799.35 | 40.21 | 201.42 | 402.09 |
| 90 | 851.76 | 882.88 | 921.78 | 43.38 | 216.90 | 433.80 | 766.73 | 3825.23 | 7648.35 | 45.21 | 226.48 | 452.12 |
| 100 | 945.36 | 976.48 | 1015.38 | 48.18 | 240.90 | 481.80 | 851.63 | 4249.73 | 8497.35 | 50.21 | 251.54 | 502.14 |

Table 5. Energy Consumed (in milliJoules)

| | FADTS | | | AMAKS | | | UPPGHA | | | MEKDA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Groups | 1 | 5 | 10 | 1 | 5 | 10 | 1 | 5 | 10 | 1 | 5 | 10 |
| 10 | 12.23 | 15.93 | 20.55 | 0.59 | 2.96 | 5.92 | 10.40 | 50.99 | 101.73 | 0.62 | 3.16 | 6.17 |
| 20 | 23.35 | 27.05 | 31.67 | 1.16 | 5.81 | 11.62 | 20.48 | 101.42 | 202.60 | 1.21 | 6.07 | 12.11 |
| 30 | 34.47 | 38.17 | 42.79 | 1.73 | 8.66 | 17.32 | 30.57 | 151.85 | 303.46 | 1.81 | 9.04 | 18.05 |
| 40 | 45.59 | 49.29 | 53.91 | 2.30 | 11.51 | 23.02 | 40.66 | 202.28 | 404.32 | 2.40 | 12.02 | 24.00 |
| 50 | 56.71 | 60.41 | 65.03 | 2.87 | 14.36 | 28.73 | 50.74 | 252.71 | 505.18 | 2.99 | 15.00 | 29.94 |
| 60 | 67.83 | 71.53 | 76.15 | 3.44 | 17.21 | 34.43 | 60.83 | 303.14 | 606.04 | 3.59 | 17.97 | 35.88 |
| 70 | 78.95 | 82.65 | 87.27 | 4.01 | 20.07 | 40.13 | 70.91 | 353.58 | 706.90 | 4.18 | 20.95 | 41.83 |
| 80 | 90.07 | 93.77 | 98.39 | 4.58 | 22.92 | 45.83 | 81.00 | 404.01 | 807.76 | 4.78 | 23.93 | 47.77 |
| 90 | 101.19 | 104.89 | 109.51 | 5.15 | 25.77 | 51.54 | 91.09 | 454.44 | 908.62 | 5.37 | 26.91 | 53.71 |
| 100 | 112.31 | 116.01 | 120.63 | 5.72 | 28.62 | 57.24 | 101.17 | 504.87 | 1009.49 | 5.97 | 29.88 | 59.65 |

*Availability:* Cluster Leaders nodes aggregate the data from nearby nodes and forward them to the Gateway. Dynamic Cluster Leader aggregation and adaptive Clustering reduce the energy overhead. MEKDA enhances security using Multi-level Authentication when an IoT device enters or exits the Cluster. The availability of the IoT devices extends by decreasing the Computation Time and Energy Consumption accomplishing secure Key distribution using

Elliptic Curve Cryptography. Elliptic Curve cryptography consumes more time than Xor but provides more security.

The randomly distributed IoT nodes group into clusters of varied *k* values using the kNN algorithm. The priority is given towards the Authentication and Key distribution process while maximizing the system availability node drain out energy due to clustering connectivity issues. The mean values with 95% confidence interval are compared in Figure 10 and Figure 11, respectively.

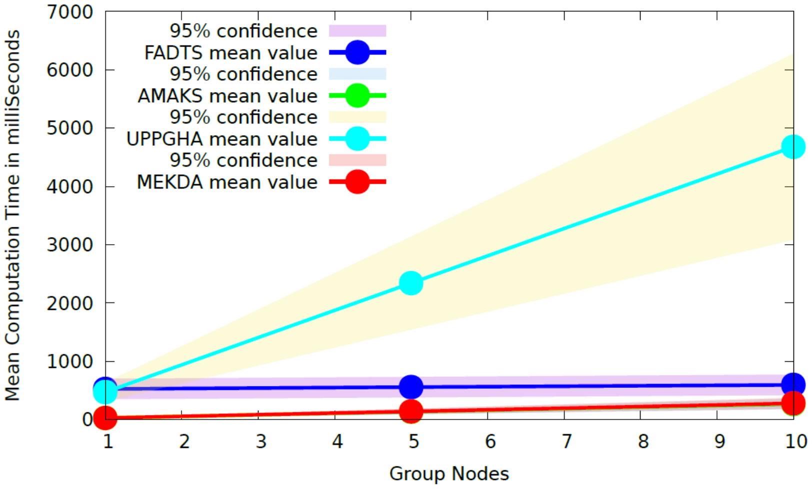

Figure 10. Mean Computation Time of the Proposed MEKDA

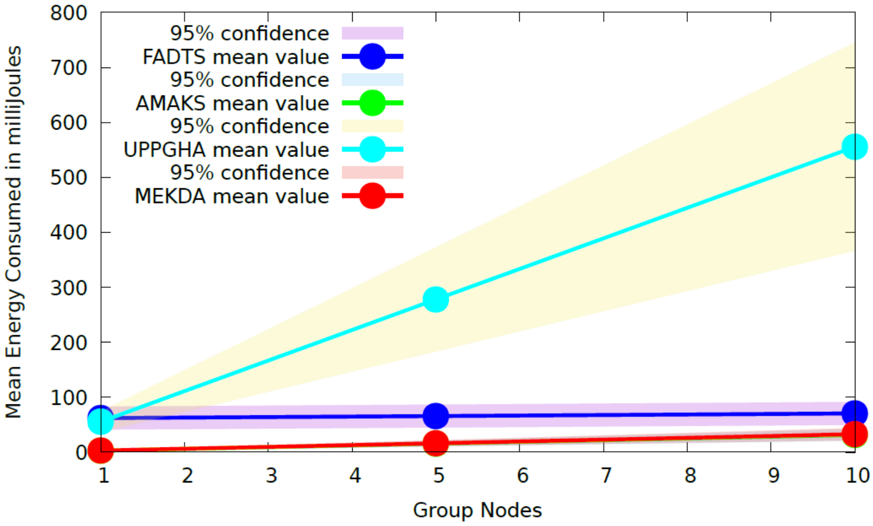

Figure 11. Mean Energy consumed by the Proposed MEKDA

The use of a single cluster leader for authentication along with the Auxiliary Leader that acts secondary node increases the lifetime of the existing cluster. The use of 256bit ECC and hashing improves the security level to 128 bits with irreversible encryption. The Cluster Leader verifies the Auxiliary Leader and Gateway verifies the Cluster Leader. Thus the MEKDA scheme enhances security by authentication and secure key distribution providing integrity, confidentiality in Cluster data communication.

## 5.1. Security Analysis

The points on the ECC-256 curve are the keys used for Cryptic functions. $G_k$ is the System Key shared to all the nodes. The Elliptic curve provides resistance from Replay, DoS, and MiTM attacks verified using SPAN Simulator. The formal verification proves the Key exchange and authentication process.

The Server selects an arbitrary secret and computes the public key $G_{k,}$ and shares it. $D_{id}$ $D_{k,}$ are the preshared keys provided during the Setup and Signing phase. Gateway computes $D_a$, $D_b$.

$$\begin{aligned} D_a &= D_{id} \oplus h(G_k \| D_k) \\ D_b &= G_k \oplus D_a \oplus D_k \\ &= G_k \oplus \{D_{id} \oplus h(G_k \| D_k)\} \oplus D_k \end{aligned}$$

The device selects two arbitrary secrets, $D_r$ and $D_{t,}$ used as an intermediary for Key Exchange and Authentication. It computes $D_{x,}$ $D_{y,}$ $D_{tid}$ to derive later by the Gateway for Authentication.

$$\begin{aligned} D_x &= D_a \oplus D_{id} \\ &= \{D_{id} \oplus h(G_k \| D_k)\} \oplus D_{id} \\ &= h(G_k \| D_k) \\ D_y &= D_x \oplus D_r \\ &= h(G_k \| D_k) \oplus D_r \\ D_{tid} &= h(D_{id} \oplus D_t \| D_r) \end{aligned}$$

Gateway receives $D_{tid}$, $D_y$, $D_a$, $D_b$, $D_{t,}$ $CL_{id'}$ and initially, verifies the Cluster Leader by verifying the known $CL_{id'}$. Then if the message is received within the possible transmission delay ($t^*$- $D_t < \delta t$), it proceeds. $G$ computes a series of statements *viz.*, $D_{k^*}$, $D_{x^*}$, $D_{id^*}$, $D_{r^*,}$ $D_{tid^*}$

using known and received variables to authenticate the device.

$$\begin{aligned} D_{k^*} &= G_k \oplus D_a \oplus D_b, \\ &= G_k \oplus \{ D_{id} \oplus h(G_k \| D_k) \} \oplus \{ G_k \oplus \{D_{id} \oplus h(G_k \| D_k)\} \oplus D_k \} \\ &= D_k \\ D_{x^*} &= h(G_k \| D_{k^*}) \\ D_{id^*} &= D_{x^*} \oplus D_a \\ &= \{h(G_k \| D_k) \} \oplus \{D_{id} \oplus h(G_k \| D_k)\} \\ &= D_{id} \\ D_{r^*} &= D_{x^*} \oplus D_y \\ &= \{ h(G_k \| D_k )\} \oplus \{h(G_k \| D_k) \oplus D_r\} \\ &= D_r \\ D_{tid^*} &= h(D_{id^*} \oplus D_t \| D_{r^*}) = h(D_{id} \oplus D_t \| D_r) \end{aligned}$$

The Gateway authenticates the device by comparing the received and derived Transient identities $D_{tid}$ and $D_{tid^*,}$ respectively. Gateway selects $D_f$, a partial session key. Gateway computes α, γ, η, μ used by the device to derive the partial keys.

$$\begin{aligned} \alpha &= D_{x^*} \oplus D_f \\ &= \{h(G_k \| D_k)\} \oplus D_f \\ \gamma &= D_{r^*} \oplus D_f \end{aligned}$$

$$= D_r \oplus D_f$$

Gateway selects an intermediary secret $D^+_{k.}$ and computes $D^+_a$, $D^+_b$ to compute η, μ, $\beta$.

$$\begin{aligned} D^+_a &= D_{id} \oplus h(G_k \| D^+_k) \\ D^+_b &= G_k \oplus D^+_a \oplus D^+_k \\ &= G_k \oplus \{ D_{id} \oplus h(G_k \| D^+_k)\} \oplus D^+_k \\ \eta &= \gamma \oplus D^+_a \\ &= \{ D_r \oplus D_f\} \oplus \{D_{id} \oplus h(G_k \| D^+_k)\} \\ \mu &= \gamma \oplus D^+_b \\ &= \{ D_r \oplus D_f\} \oplus \{ G_k \oplus D_{id} \oplus h(G_k \| D^+_k) \oplus D^+_k\} \\ \beta &= h(D_x \| D_r \| D_f \| \eta \| \mu) \end{aligned}$$

The session key is $s_{k*}=h(D_{id}\|D_r\|D_f\|D_x)$. Gateway sends {α, β, η, μ}=$\{h(G_k\|D_k)\oplus D_f, h(D_x\|D_r\|D_f\|\eta\|\mu), D_r\oplus D_f\oplus D_{id} \oplus h(G_k\|D^+_k), D_r\oplus D_f\oplus G_k \oplus D_{id} \oplus h(G_k\|D^+_k)\oplus D^+_k\}$ to the Device *via* the Cluster Head and the Auxiliary Cluster Head. The device, on receiving the required parameters, computes $D_{f*}$, $\beta*$ derives the session key.

$$\begin{aligned} D_{f*} &= D_x \oplus \alpha \\ &= \{h(G_k\|D_k)\} \oplus \{h(G_k\|D_k) \oplus D_f\} \\ &= D_f \\ \beta* &= h(D_x\|D_r\|D_{f*}\| \eta \| \mu) \end{aligned}$$

The Device validates the message authentication by verifying recieved $\beta$ and derived $\beta*$.

It computes γ, $D^+_a$, and $D^+_b$.

$$\begin{aligned} \gamma &= D_r \oplus D_{f*} \\ &= D_r \oplus D_f \\ D^+_a &= \gamma \oplus \eta \\ &= \{ D_r \oplus D_f\} \oplus \{D_r \oplus D_f \oplus D_{id} \oplus h(G_k\|D^+_k)\} \\ &= D_{id} \oplus h(G_k\|D^+_k) \\ D^+_b &= \gamma \oplus \mu \\ &= \{G_k \oplus D_{id}\} \oplus \{ h(G_k\|D^+_k) \oplus D^+_k\} \\ s_{k*} &= h(D_{id}\|D_r\|D_f\|D_x) \end{aligned}$$

Thus, the device receives the session key without revealing any secrets to the intermediary Cluster Head of the Cluster Auxiliary Leader. When the Cluster Leader resource is depleted, the Auxiliary Leader node acts as the prospective leader.

Table 6 compares the computation functionality of MEKDA with FADTS [1], AMAKS [27], and UPPGHA [28]. FADTS requires Scalar Multiplications with Hashing, AMAKS uses XoR and Hashing function, UPPGHA uses Exponential function. MEKDA uses XoR and scalar multiplication leads to a reduction in time complexity, and Hashing provides integrity. The XoR reduces the time complexity but the reversibility breaches security; the use of scalar elliptic curve multiplication enhances the security by concealing the secrets in the network. The messages received by an adversary cannot retrieve the keys due to DDH and CDH hypothesis being unsolvable in polynomial time.

Table 6. Comparison of Computation

| | FADTS | AMAKS | UPPGHA | MEKDA |
|---|---|---|---|---|
| Device | $5T_m + 2T_h$ | $6T_x + 3T_h$ | $5T_e$ | $7T_x + 3T_h + T_m$ |
| Gateway | $5T_m + 4T_h$ | $11T_x + 5T_h$ | $5T_e$ | $12T_x + 5T_h + T_m$ |

Time of $T_m$: Scalar Multiplication; $T_e$: Modular Exponentiation; $T_x$: XoR; $T_h$: Hashing;

Thus MEKDA shows better performance than other algorithms due to the use of ECC and reduced communications but the bandwidth. However, the real-time implementations may vary due to the heterogeneous IoT device characteristics.

## 6. CONCLUSIONS

In an IoT system, the deployed devices generate a significant number of sensitive data. The Authentication of a wide range of devices deployed in the IoT environment, and Key Distribution among them, is a security challenge. The security of IoT depends mainly on the secrecy of the Cluster Key. The Multi-level Elliptic Curve Cryptography-based Key Distribution and Authentication in the Internet of Things enhances the security of the System by Multi-level Authentication of the IoT devices when enters or exits the Cluster. The availability of the IoT devices extended by decreased Computation Time and Energy Consumption accomplishing secure Key distribution using Elliptic Curve Cryptography. The Computation Time and Energy Consumption of MEKDA are two times that of FADTS and 16 times that of UPPGHA. The decrease in computation time and Energy Consumption extends the availability of the IoT devices by accomplishing Key distribution using ECC and Hashing with Xor enhancing security by authentication providing confidentiality and integrity. The performance of MEKDA may vary in a hardware setup with a large number of real-time IoT devices. Hybrid Authentication can further improve performance.

### CONFLICTS OF INTEREST

The authors declare no conflict of interest.

### REFERENCES

[1] J. Cao, P. Yu, M. Ma, and W. Gao, “Fast Authentication and Data Transfer Scheme for Massive NB-IoT Devices in 3GPP 5G Network,” *IEEE Journal on Internet of Things,* vol. 6, no. 2, pp. 1561–1575, 2019.

[2] U. Khadam, M. M. Iqbal, M. Alruily, M. A. Al Ghamdi, M. Ramzan, and S. H. Almotiri, “Text Data Security and Privacy in the Internet of Things: Threats, Challenges, and Future Directions,” *Hindawi Journal on Wireless Communications and Mobile Computing,* vol. 2020, pp.1-15, 2020.

[3] J. Dizdarevi´c, F. Carpio, A. Jukan, and X. Masip-Bruin, “A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration,” *ACM Journal on Computing Surveys,* vol. 51, no. 6, pp. 1–30, 2019.

[4] A. Mansour, K. M. Malik, A. Alkaff, and H. Kanaan, “ALMS: Asymmetric Lightweight Centralized Group Key Management Protocol for VANETs,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 22, no. 3, pp. 1663 - 1678, 2021.

[5] S. Dey and A. Hossain, “Session-Key Establishment and Authentication in a Smart Home Network Using Public Key Cryptography,” *IEEE Sensors Letters,* vol. 3, no. 4, pp. 1–4, 2019.

[6] Y. Aydin, G. K. Kurt, E. Ozdemir, and H. Yanikomeroglu, “A Flexible and Lightweight Group Authentication Scheme,” *IEEE Journal on Internet of Things,* vol. 7, no. 10, pp. 10277 - 10287, 2020.

[7] J. Cao, M. Ma, and H. Li, “G2RHA: Group-to-Route Handover Authentication Scheme for Mobile Relays in LTE-a High-Speed Rail Networks,” *IEEE Transactions on Vehicular Technology,* vol. 66, no. 11, pp. 9689– 9701, 2017.

[8] J. Cao, P. Yu, X. Xiang, M. Ma, and H. Li, “Anti-Quantum Fast Authentication and Data Transmission Scheme for Massive Devices in 5G NB-IoT System,” *IEEE Journal on Internet of Things,* vol. 6, no. 6, pp. 9794–9805, 2019.

[9] S. Kavianpour, B. Shanmugam, S. Azam, M. Zamani, G. Narayana Samy, and F. De Boer, “A Systematic Literature Review of Authentication in Internet of Things for Heterogeneous Devices,” *Hindawi Journal of Computer Networks and Communications,* vol. 2019, pp.1-14, 2019.

[10] X. Jiang, M. Lora, and S. Chattopadhyay, “An Experimental Analysis of Security Vulnerabilities in Industrial IoT Devices,” *ACM Transactions on Internet Technology,* vol. 20, no. 2, pp. 16:1–16:24, 2020.

[11] S. C. Lin, C. Y. Wen, and W. A. Sethares, “Two-Tier Device-based Authentication Protocol Against PUEA Attacks for IoT Applications,” *IEEE Transactions on Signal and Information Processing over Networks,* vol. 4, no. 1, pp. 33–47, 2018.

[12] M. Dammak, S. M. Senouci, M. A. Messous, M. H. Elhdhili, and C. Gransart, “Decentralized Lightweight Group Key Management for Dynamic Access Control in IoT Environments,” *IEEE Transactions on Network and Service Management,* vol. 17, no. 3, pp. 1742 - 1757, 2020.

[13] P. Xu, S. He, W. Wang, W. Susilo, and H. Jin, “Lightweight Searchable Public-Key Encryption for Cloud-Assisted Wireless Sensor Networks,” *IEEE Transactions on Industrial Informatics,* vol. 14, no. 8, pp. 3712– 3723, 2018.

[14] Q. Cheng, C. Hsu, and L. Harn, “Lightweight Noninteractive Membership Authentication and Group Key Establishment for WSNs,” *Hindawi Journal on Mathematical Problems in Engineering,* vol. 2020, pp. 1–9, 2020.

[15] J. Zhang, H. Zhong, J. Cui, Y. Xu, and L. Liu, “An Extensible and Effective Anonymous Batch Authentication Scheme for Smart Vehicular Networks,” *IEEE Journal on Internet of Things,* vol. 7, no. 4, pp. 3462– 3473, 2020.

[16] Z. Y. Wu, “Group-Oriented Cryptosystem for Personal Health Records Exchange and Sharing,” *IEEE Access,* vol. 7, pp. 146 495–146 505, 2019.

[17] J. Qiu, K. Fan, K. Zhang, Q. Pan, H. Li, and Y. Yang, “An Efficient Multi-Message and Multi-Receiver Signcryption Scheme for Heterogeneous Smart Mobile IoT,” *IEEE Access,* vol. 7, pp. 180 205–180 217, 2019.

[18] H. Y. Chien, “Group-Oriented Range-Bound Key Agreement for Internet of Things Scenarios,” *IEEE Journal on Internet of Things,* vol. 5, no. 3, pp. 1890–1903, 2018.

[19] A. K. Das, S. Member, M. Wazid, and Y. Park, “Provably Secure ECC based Device Access Control and Key Agreement Protocol for IoT Environment,” *IEEE Access,* vol. 7, pp. 55 382–55 397, 2019.

[20] C. Esposito, M. Ficco, A. Castiglione, F. Palmieri, and A. De Santis, “Distributed Group Key Management for Event Notification Confidentiality among Sensors,” *IEEE Transactions on Dependable and Secure Computing,* vol. 17, no. 3, pp. 566 - 580, 2020.

[21] D. Liu, X. Liu, H. Zhang, H. Yu, W. Wang, L. Ma, J. Chen, and D. Li, “Research on End-to-End Security Authentication Protocol of NB-IoT for Smart Grid based on Physical Unclonable Function,” *in Proceedings of the Eleventh IEEE International Conference on Communication Software and Networks,* pp. 239–244, 2019.

[22] Z. Mahmood, A. Ullah, and H. Ning, “Distributed Multiparty Key Management for Efficient Authentication in the Internet of Things,” *IEEE Access,* vol. 6, pp. 29 460–29 473, 2018.

[23] Z. Gu, H. Chen, P. Xu, Y. Li, and B. Vucetic, “Physical Layer Authentication for Non-Coherent Massive SIMO-Enabled Industrial IoT Communications,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3722–3733, 2020.

[24] Z. Qikun, G. Yong, T. Yu-An, Z. Quanxin, and W. Ruifang, “A Dynamic and Cross-Domain Authentication Asymmetric Group Key Agreement in Telemedicine Application,” *IEEE Access,* vol. 6, pp. 24 064–24 074, 2018.

[25] P. Vijayakumar, M. S. Obaidat, M. Azees, S. H. Islam, and N. Kumar, “Efficient and Secure Anonymous Authentication with Location Privacy for IoT-based WBANs,” *IEEE Transactions on Industrial Informatics,* vol. 16, no. 4, pp. 2603–2611, 2020.

[26] C. S. Park and W. S. Park, “A Group-Oriented DTLS Handshake for Secure IoT Applications,” *IEEE Transactions on Automation Science and Engineering,* vol. 15, no. 4, pp. 1920–1929, 2018.

[27] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta, and K. K. R. Choo, "Anonymous Mutual Authentication and Key Agreement Scheme for Wearable Sensors in Wireless Body Area Networks," *Elsevier Journal of Computer Networks,* vol. 129, pp. 429–443, 2017.

[28] J. Cao, H. Li, M. Ma, and F. Li, "UPPGHA: Uniform Privacy Preservation Group Handover Authentication Mechanism for mMTC in LTE-A Networks," *Hindawi Journal of Security and Communication Networks,* vol. 2018, pp.1-16, 2018.

[29] M. G. Padmashree, S. Khanum, J.S. Arunalatha, and K. R. Venugopal, "SIRLC: Secure Information Retrieval using Lightweight Cryptography in HIoT," *in Proceedings of the IEEE International Technical Conference of Region 10 (TENCON 2019),* pp. 169-173, 2019.

[30] M. G. Padmashree, Ranjitha, J. S. Arunalatha, and K. R. Venugopal, "CKDAC: Cluster-Key Distribution and Access Control for Secure Communication in IoT," *in Proceedings of the Seventh IEEE Uttar Pradesh Section International Conference on Electrical,* Electronics and Computer Engineering (UPCON 2020), pp. 1-6, 2020.

[31] M. G. Padmashree, J. S. Arunalatha, and K. R. Venugopal, " HPAKE: Hybrid Precocious Authentication and Key Establishment in IoT," *in Proceedings of the FiftyThird IEEE International Carnahan Conference on Security Technology (ICCST 2019),* pp. 129-134, 2019.

[32] M. G. Padmashree, S. Khanum, J. S. Arunalatha, and K. R. Venugopal, "ETPAC: ECC based Trauma Plight Access Control for Healthcare Internet of Things," *Springer International Journal of Information Technology*, vol. 13, no. 4, pp. 1481–1494, 2021.

[33] B. Kapito1, M. Nyirenda1, and H. Kim, "Privacy-Preserving Machine Authenticated Key Agreement for Internet of Things," *International Journal of Computer Networks & Communications (IJCNC),* vol. 13, no. 2, pp. 99-120, 2021.

[34] Lakshmi M and Prashanth C R, "Designing an Energy Efficient Clustering in Heterogeneous Wireless Sensor Network," *International Journal of Computer Networks & Communications (IJCNC),* vol. 13, no. 1, pp. 75-92, 2021.

## AUTHORS

**Padmashree M. G.** received the B.E. Degree in Computer Science & Engineering from JNNCE, Shivamogga, Kuvempu University, Karnataka, India, in 1998 and the M.Tech. Degree in Computer Science & Engineering from RVCE, Bengaluru, Visvesvaraya Technological University, Karnataka, in 2011. She is currently pursuing a Ph.D. degree in Computer Science and Engineering at Bangalore University, Bengaluru, India. She has published 6 articles in refereed International Journals and Conferences. Her research interest includes Scheduling Techniques in Operating Systems, Cryptography, Security in the Internet of Things.

**Mallikarjun** received the B.E. Degree in Computer Science & Engineering from Visvesvaraya Technological University, India, and the M.Tech. Degree in Information Technology from Bangalore University, India.

**Arunalatha J S** is a Professor in the Department of Computer Science and Engineering at University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India. She obtained her Bachelor of Engineering in Computer Science and Engineering, from PES College of Engineering, Mandya, Mysore University, and she received her Master's degree in Computer Science and Engineering from Bangalore University, Bangalore. She pursued her Ph.D. in the area of Biometrics and has published 17 articles in refereed International Journals and Conferences. Her research interest is in Biometrics, Image Processing, IoT, Big Data Analytics, and Web Mining.

**Venugopal K R** is currently the Vice-Chancellor of Bangalore University, Bangalore. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering and received his Master's degree in Computer Science and Automation from Indian Institute of Science Bangalore. He was awarded a Ph.D. in Economics from Bangalore University, and Ph.D. in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science, and Journalism. He has authored and edited 77 books on Computer Science and Economics. He has over 980 research papers to his credit. His research interests include Computer Networks, Wireless Sensor Networks, Parallel and Distributed Systems, Digital Signal Processing, Data Mining, IoT, and Cloud Computing. He received IEEE Fellow and ACM Distinguished Educator award from the USA for his outstanding contributions to Computer Science and Engineering.