Journal of Harbin Engineering University Vol 44 No. 5
ISSN: 1006-7043 May 2023

A Safe and Power-Proficient Data Transmission Framework for the Internet of Things

K. Pradeepa[1*], Dr. M. Parveen[2]
[1]Assistant Professor, Department of Computer Science, Cauvery College for Women (Autonomous), Trichy.
[2]Professor and Head, Department of Information Technology, Cauvery College for Women (Autonomous), Trichy. Affiliated to Bharathidasan University

Abstract:

The idea of the Internet of Things (IoT) has recently received much attention from businesses and academics. In the IoT, a base station receives data from millions of sensor devices and processes it before using it to build various smart systems, such as the smart grid, smart city, and smart healthcare. A secure link must be established between the base station and sensor devices to ensure the accuracy of the data gathered. The data analysis findings will be erroneous and cause more severe harm if the collected data is corrupt. Additionally, due to their extremely low-power computational processors, these IoT devices have a very low level of interactivity. These devices perceive their surroundings, produce data, and transfer it to the base station through intermediary devices. The data is delivered to the base station using some routing algorithms with the aim of low power consumption. Power efficiency should be considered a crucial performance metric when utilizing low-power IoT devices to create a routing algorithm. Therefore, this paper presented a safe and energy-efficient data transmission framework (SE-DTF) for IoT. This framework consists of three phases. The first phase is a public and secret key with a token sharing (IoT-PSKTS) algorithm, which is used to prevent key leakage in the IoT. The second phase focuses on low power consumption using the Hierarchical Fuzzy Logic Clustering (HFLC) algorithm and Minimum Power Consumption Routing (MPCR) algorithm. The third phase focuses on safe data transfer, employing two-tier cryptography with ciphertext shifting, token-based access control, and HMAC-SHA1 signature. The experimental findings demonstrate how securely the IoT-PSKTS algorithm can share a public and a secret key with a token. It also demonstrates that the MPCR with the HFLC algorithm outperforms existing algorithms regarding throughput, power utilization, and packet delivery ratio. Additionally, it demonstrates that the two-tier cryptography technique uses less energy and requires less computation time for encryption and decryption than other cryptography techniques.

Keywords: Key sharing, clustering, routing, cryptography, access control, signature, and ciphertext shifting

1 Introduction:

Several sectors, including smart homes, smart cities, medical services, industrial automation, disaster management, agriculture, etc., use the IoT in various applications. The IoT network often uses wirelessly connected devices to carry out a certain smart activity [1]. The IoT network differs from the conventional wireless sensor network (WSN) in that its devices dynamically connect to the internet and cooperate with other related devices to perform tasks [2]. Like the traditional WSN network, IoT devices regularly transmit the sensed data to the base station (BS). The administrator receives the sensed data after the base station sends it. In addition, ioT utility devices frequently use transceivers, batteries, and microprocessors to carry out essential operations [3].

This characteristic allows the IoT network to be used for various purposes, such as tracking, remote access, and automation. The IoT network has many uses and is more well-known in almost every technological area. For instance, crucial applications include monitoring industrial processes, healthcare, battlefields, and smart city traffic [4]. The appliance is used for remote access and is automated, and it incorporates the Smart Grid, Smart Hospital, Smart Home, and more.

IoT systems are used for various sectors, such as smart farming to boost production, smart traffic monitoring, smart health monitoring,

disaster detection systems, and water control, surveillance, and leak prevention systems. The primary requirement for completing all applications is to collect and transfer data. Any IoT device's limited communication capabilities could only transfer data up a certain distance. The other intermediary devices must cooperate to transmit data from the IoT device to the base station. The IoT device serves as both a source and a relay. Low latency, low power, and high security should all be used to transfer data to the base station [5]. As a result, this paper presented a framework for Secure and Energy-Efficient Data Transmission (SE-DTF). This framework comprises three phases.

IoT-PSKTS, an algorithm that uses public and secret keys with token sharing, avoids key leakage in the initial phase. IoT network is made up of three entities: (Administrator), (IoT Base Station), and (IoT Devices). The administrator is in charge of field monitoring, and with the help of IoT, he may do it from any location whenever. IoT devices watch the environment and produce perceived data. Then, through an IoT base station, it transmits sensed data to the administrator. An IoT network controller is an IoT base station. After gathering sensed data, it sends the data to the administrator. IoT devices have constrained resources, including limited energy and coverage areas.

Each IoT device is set at different positions throughout the field to measure temperature, sound, vibration, pressure, and other variables. It must be encrypted to exchange this sensed data with the admin safely. Public and secret keys are required to encrypt this sensed data. Access control can also guarantee that IoT devices have appropriate network access. Consequently, the admin creates the access control token for each IoT device with these keys. The IoT-PSKTS algorithm was designed to safely share these keys with a token to each IoT device.

The second phase uses the Hierarchical Fuzzy Logic Clustering (HFLC) algorithm and the Minimum Power Consumption Routing (MPCR) algorithm to reduce energy consumption. We may reduce the quantity of data broadcast to the base station by utilizing the MPCR algorithm to aggregate the data in the cluster head, lower communication expenses, and conserve battery power. In addition, the MPCR algorithm considerably lowers the network overheads for maintaining the optimum route.

The implementation of cluster generation is based on hierarchical clustering, which produces a cluster hierarchy by frequently combining several small clusters into a big one or breaking a big cluster into small clusters. Implementations of cluster head formation that use fuzzy logic are machine learning techniques that can be used in uncertain applications. For example, building clusters in the IoT may not be appropriate if it uses predefined rules because the competency relies on overlapping parameters like power, the number of devices, the distance between devices and base stations, etc. Therefore, using fuzzy logic to address the uncertainty in choosing the cluster head is appropriate.

The third phase focuses on safe data transfer, employing two-tier cryptography with ciphertext shifting, token-based access control, and HMAC-SHA1 signature. In Tier-1, data sensed by an IoT device is encrypted using a secret key with a ciphertext shifting-based encryption technique. In Tier 2, the ciphertext received from each IoT device by the IoT base station is encrypted using the public key with a ciphertext shifting-based cryptography technique. Additionally, a token-based access control technique is utilized to let merely the administrator who has the token of the sender IoT device access the received ciphertext. Additionally, the HMAC-SHA1 signature is used to verify whether the received ciphertext is secure.

The remainder of this paper is structured in the following manner. Section 2 examined the relevant literature. Section 3 provided more information on the Secure and Energy-efficient Data Transmission Framework (SE-DTF). Section 4 covered the performance of the suggested framework. Finally, section 5 provides a conclusion to this paper.

2 Related works:

This section highlights the associated works on existing access control, routing, cryptography, and secret-sharing techniques.

Tang et al. [6] proposed an efficient secret key distributing method for MIMO (multiple-input-multiple-output) applications of IoT that could recognize secret key sharing and enable concurrent interaction. The authentic receiver's SVD (single value corruption) MIMO channel generated several tiny active or inactive channels mixed with the secret key information. As a result, the authorized recipient can precisely locate the shared key and erase the communication details by using optional low-key BER (bit error rate).

A novel method of recording incorporated information in IoTs using a secret distribution technique and (t, n)-threshold in the cloud was discussed by Farhadi et al. [7]. This method separated the real data into t chunks, and each was considered a share. This method can add novel data or remove a piece of the real data devoid of altering shares, and it can be detected when CSPs discover a mistake by providing the incorrect share.

With the help of the power load-balancing technique, Maheswar et al. [8] established a cluster-based backpressure routing (CBPR) method to extend the network's lifespan and enhance data transfer dependability. For each cluster of the sensing device, the CBPR method chooses the cluster head with the highest power level. In addition to employing the Backpressure scheduling machine for data packet queuing and the route chosen, the CBPR routing technique also uses a more powerful data integration method to defeat and prevent the flow of redundancy packets in the network. Furthermore, it enables it to choose the next-hop sensor device using the queue size value of the sensor devices.

Raj Kumar et al. [9] devised a routing protocol based on clustering for WSN-IoT. This protocol primarily consists of three phases. The first phase is establishing the network's transmission zone. The zone cluster would be formed during the second phase, and the zone head would be selected during the final phase according to power and location. The base station and zone head broadcast might be multi-hop or single-hop, depending on the situation.

For Low-Power and Lossy Networks, Sankar et al. [10] provided a multi-layered clustering-based routing method that divided the network into rings of comparable size. A ring is divided into perfectly equal clusters using the intra-ring clustering technique, and the best path for data transfer is chosen using inter-cluster routing. It prolongs network life and improves the ratio of packet delivery.

A lightweight searchable attribute-based encryption (ABE) scheme was presented by Zhang et al. (that is, LSABE). IoT device computing costs could be significantly reduced if their technology is used because data users can search for multiple keywords. Furthermore, to efficiently make and handle secret or public keys in the disseminated IoT surroundings, the authors modified the LSABE system to a multi-authority scenario. Finally, the authors concluded that, compared to other earlier systems, their systems could considerably keep computational competence and reduce the computational expenditure of IoT gadgets.

A hybrid encryption technique was created by Yousefi et al. [12] to reduce security concerns, enhance encryption speed, and need less computing complexity. The dependability of the data, confidentiality, and non-rejection in data transfer for the IoT are the driving forces behind this hybrid algorithm. Last but not least, the encryption method created by MATLAB software was tested for speed and security compared to the standard encryption technique.

A strong secret key exchange and joint authentication technique was offered by Choudhary et al. [13] to address the weaknesses of the earlier approaches. The authors provided secure joint authentication and secret key swap among several entities to prevent unauthorized access by utilizing various cryptographic functions, such as hashing, ciphering, and others. As a result, the authors concluded that their protocol is more

energy efficient than conventional systems (cheaper communication and computing) and more resilient to attacks.

IoT access control was developed and implemented in a novel way by Schuster et al. [14]. Their key innovation is the introduction of "environmental situation oracles" (ESOs) as premium items inside the IoT ecosystem. An ESO summarizes the steps taken to sense or infers a situation. Situational constraints may be imposed by IoT access-control structures using ESOs; however, ESOs and structures are uninformed of one another's execution details. Multiple access-control mechanisms could be applied to a single ESO throughout the ecosystem. Since ESOs summarize responsive device-access rights, this also sustains dependable enforcement of general policies and reduces the overprivileged. ESOs might be organized by any IoT software layer tier employing access control. The authors executed prototype ESOs for the IoT resource tier and IoT Web services using the IoTivity structure and Passport middleware, respectively.

According to Terkawi et al. [15], the importance of IoT access control will be highlighted in the future as IoT device numbers increase and business models improve. Therefore, the traditional access control methods will eventually give way to something completely new that is better appropriate for the IoT, like their suggested model that combines authentication and verification. Additionally, implementing regulations would allow for more flexibility and decentralized administration of distributed resources, a key objective in certified combined surroundings.

3 Secure and Energy-efficient Data Transmission Framework (SE-DTF):

This section describes the proposed Secure and Energy-efficient Data Transmission Framework (SE-DTF). Since this framework focused on reducing energy consumption while routing and securing data transmission in IoT. Figure 1 demonstrates the architecture of the SE-DTF framework.

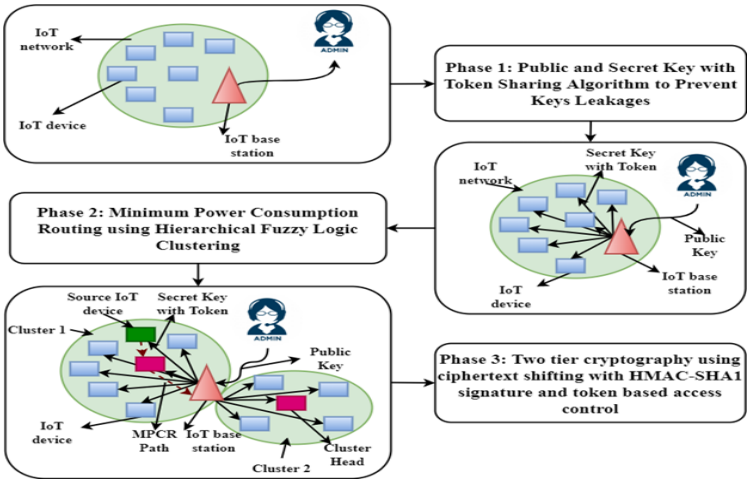

Figure 1: Architecture of SE-DTF framework

This framework has 3 phases. The first phase is public and secret keys with a token sharing (IoT-PSKTS) algorithm, which is used to prevent key leakage. This algorithm enables secure communication over an untrusted IoT network by setting up shared keys between two or more parties. The second phase focuses on low power consumption using the Hierarchical Fuzzy Logic Clustering (HFLC) algorithm and Minimum Power Consumption Routing (MPCR) algorithm. Finally, the third phase focuses on safe data transfer, employing two-tier cryptography with ciphertext shifting, token-based access control, and HMAC-SHA1 signature. The following sections discuss each phase.

3.1 Phase 1: Public and secret keys with token sharing (IoT-PSKTS) algorithm:

By enabling the interchange of encryption and decryption keys among two entities, the cryptographic approach known as "key sharing" makes it possible to use a certain algorithm. Sharing any keys or other information is necessary to build a safe communication channel where no one else may access a copy. Hence, it is known as the "key exchange problem." In the past, symmetric-key cryptography, which predates the development of public-key cryptography (asymmetrical cryptography), used a solitary key for message encryption and decryption. Two parties should initially exchange the secret key to be capable of encryption and decryption interactions. Only then can they be capable of interacting covertly. This process is known as key exchange.

Symmetric cryptography, also known as single-key cryptography, presents a significant challenge. Securing a confidential key via authorized couriers or other secure communication channels necessitates securing it. For example, suppose a secure initial key exchange cannot be established. In that case, two parties may risk having their messages intercepted and decrypted through a third party who has gained access to the key during the initial swap, compromising their ability to communicate securely.

In public-key cryptography, messages are encrypted using one key and decrypted using another. This system involves using two keys: the public key and the private key. The key choice - public or private - for encrypting or decrypting messages depends on the cryptographic algorithm used. For instance, in RSA, the private key is used for message decryption, while in the Digital Signature Algorithm (DSA), it is used for message authentication. Only the owner has exclusive access to the private key, while the public key can be shared openly or transmitted through insecure communication channels.

The process known as the Diffie-Hellman key exchange enables the exchange of encryption keys without any risk to the confidentiality of the encrypted messages. To enable the other party to use the key for encrypting messages and sending back the resulting cipher text, one party must exchange the key with the other. The cipher text received can only be decrypted using the private key, which is kept confidential. Unlike a symmetrical key exchange, the Diffie-Hellman key exchange does not compromise sensitive information security, ensuring the transmitted messages' privacy and authenticity.

Within the realm of IoT, an administrator creates a set of security credentials for encryption, decryption, and access control purposes. These security credentials consist of the following:

- A public key encrypts data sent to the IoT base station.
- A secret key unique to each IoT device is used to encrypt data sent from that device.
- A token specific to each IoT device is used for access control.
- A private key that is kept confidential and is used for decrypting data for the administrator's use.

Each IoT device needed a secret key to encrypt sensed data, and each IoT device needed a token for access control. In addition, the IoT base station needed a public key for encryption. Therefore admin needed to share these keys with the base station and each IoT device. While sharing these keys to avoid key leakage, the admin used the first phase of the SE-DTF framework, namely public and secret keys, with a token sharing (IoT-PSKTS) algorithm.

During the IoT-PSKTS phase, the security administrator transforms the token, secret, and public key into n shares. These n shares are then transmitted to the IoT base station. Using these shares, the IoT base station can reconstruct the public key, secret key, and token, even if it receives fewer than the full n shares. The IoT base station then converts each IoT device's secret key and token into n shares, which are transmitted to the corresponding device. After receiving any

number of shares that is less than n, each IoT device can reconstruct its secret key and token.

3.2 Phase 2: Minimum Power Consumption Routing (MPCR) algorithm using Hierarchical Fuzzy Logic Clustering (HFLC) algorithm:

In an IoT network, devices could smartly communicate with one another. Devices perceive the value of the deployed environment. After sensing, the devices plan to relay their detected data to the base station. However, in a large-scale IoT network, many devices are located distant from the base station, making it impossible for them to transfer data straightforwardly to the base station. Therefore, a routing approach is required. To solve this issue, routing is discovering a suitable path from a source device to a base station.

The device's energy transfers the sensed data via an appropriate path from the source IoT device to the base station. Energy restrictions are one of the greatest problems with IoT routing. Discovering the most direct path is accomplished through existing routing methods, such as Ad hoc On-demand Distance Vector (AODV) and Dynamic Source Routing (DSR). Unfortunately, these approaches fail to account for a node's energy consumption. As a result, a certain device might be picked up repeatedly, drastically reducing its lifespan.

Metrics are utilized to assess a link's or path's quality. The most commonly utilized metric is hop count, which chooses the path with the least number of hops. Most current routing protocols employ hop count as their route selection metric to discover the quickest route between sender devices to the base station. However, this is frequently not the best option because long-distance links are particularly prone to loss, and not all links are created equally in quality. But, using just hop count as a routing metric is inappropriate for the IoT since it will not select that route if a better route with more bandwidth and usable device residual energy is available. These routing protocols can also not significantly reduce the power expenditure of IoT devices. As a result, developing effective routing algorithms should require energy awareness.

Additionally, devices cannot interact directly with each other if they are far apart because of the limits of signal transmission range, and trying to do so uses extra energy. The IoT network requires to be clustered to solve this issue. Clustering is one of the design strategies for effectively managing network power consumption since it reduces the number of devices participating in long-distance communication with the base station and distributes the power usage among devices within the network. Furthermore, clustering divides devices into discrete groups. Thus, creating effective routing algorithms requires energy awareness and clustering awareness. Cluster-based routing's fundamental objective is to maintain device power utilization.

The IoT network's routing algorithm allows any originating IoT device to transmit data to the base station if there is a one-hop connection between the device and the station. In cases where this is impossible, the data is relayed from the originating device to its neighbor, who then transmits it to the base station using additional intermediate devices as necessary. Numerous routing algorithms dictate how data is transmitted from an IoT device to the base station. One particularly well-regarded routing algorithm is the cluster-based approach, which offers distinct advantages when used in low-power IoT networks.

Cluster-based routing algorithms significantly reduce the number of data packets transmitted to the base station. However, as IoT devices become increasingly ubiquitous in networks, the base station's data collection will inevitably increase network overhead and result in redundant data being gathered for decision-making. By employing a cluster-based routing algorithm, the cluster head is responsible for gathering data from its members, combining it, and transmitting a single aggregated data packet to the base station. It greatly reduces the data packets the base station needs to collect, streamlining the process and optimizing network performance.

Scientists and academics are interested in clustering in IoT networks because of their advantages for creating an energy-efficient network. A cluster head is selected during the clustering process, and other members are provided with data on the cluster head. It is type-1 for cluster formation, where the members who want to be cluster members can join the cluster with the cluster leader. A cluster head is taken after the cluster is built; it is type-2 for creating the cluster.

The administrator applied the second phase of the SE-DTF framework, namely the Minimum Power Consumption Routing (MPCR) algorithm using the Hierarchical Fuzzy Logic Clustering (HFLC) algorithm for the IoT network. This HFLC algorithm generates clusters using type-2. The creation of clusters is accomplished through a hierarchical clustering method. This technique constructs a hierarchical structure of clusters by amalgamating numerous smaller clusters into one larger cluster or partitioning a larger cluster into smaller clusters. Cluster head selection is implemented based on the Fuzzy Logic technique. One of the computational intelligence methods used in uncertain applications is fuzzy logic.

First, the administrator creates an IoT network using IoT devices and a base station. Then the administrator clusters the IoT network for routing purposes using the HFLC algorithm. Using intermediate cluster heads, a base station discovers all routes between each cluster head. It ranks all routes to each cluster head from shortest path to longest path. Each cluster head then reports its route details. Then, each cluster head stores its sorted route details in its routes list.

Whenever an IoT device (called a "cluster member") detects an event, it produces a packet that is transmitted to its respective cluster head. The cluster head then accumulates all the received packets until the total number of packets reaches the predetermined packet aggregation threshold. Once the number of received packets matches the aggregation threshold, the cluster head forwards the combined packets to the following device (another cluster head or the base station) along the shortest possible path. If the next device is a cluster head, it repeats the above step. But the next device is the base station; It sends the aggregated packet to the administrator. After receiving the aggregated packet, the administrator extracts each packet from the aggregated packet. If any failure occurs from any shortest path, all cluster heads take the next shortest path from their path list. The main advantages of this algorithm are less battery usage and successful packet delivery.

The MPCR algorithm aims to enhance the lifetime of the IoT network. Hence this work proposes a Hierarchical Fuzzy Logic Clustering (HFLC) algorithm to generate minimum power consumption routing, which leads to network lifetime extension. In the HFLC algorithm, cluster formation is based on hierarchical clustering. An unsupervised clustering algorithm called hierarchical clustering creates clusters with a top-down hierarchy of importance. For example, our hard drive's folders and data are hierarchically arranged. The algorithm groups comparable objects into clusters. An endpoint is a group of clusters, each of which differs from the others but typically contains similar objects. Hierarchical clustering merges multiple smaller clusters into one larger cluster or splits a larger cluster into smaller ones, resulting in a cluster hierarchy. Two types of hierarchical clustering are Aggregation Hierarchical Clustering and Extractive Hierarchical Clustering.

Aggregate hierarchical clustering is a "bottom-up" method, whereas divisive hierarchical clustering is a "top-down" one. Therefore, the proposed HFLC algorithm is a "top-down" clustering approach; in the HFLC algorithm, the administrator assigns all IoT devices to a single cluster and divides the cluster into two less equal clusters. Finally, the administrator repeatedly divides each cluster until there is a cluster for each IoT device.

The HFLC algorithm also implements cluster head selection based on the Fuzzy Logic technique. One of the computational intelligence methods used in uncertain applications is fuzzy logic. Efficient cluster head selection is achieved by utilizing three fuzzy parameters, including the IoT

device's present power level, the distance between the IoT device and the base station, and the distance between the cluster members and the IoT device.

3.3 Phase-3: Two-tier cryptography using ciphertext shifting with HMAC-SHA1 signature and token-based access control

This section discusses how IoT devices can securely transmit data to an administrator via an IoT base station. This communication must fulfill confidentiality, authentication, integrity, anonymity, and access control. Confidentiality means keeping messages confidential from others except for the sender's IoT device and the administrator. Authentication guarantees that valid IoT devices can send data to the administrator via the IoT base station. Integrity ensures that illegitimate entities do not alter data on IoT devices. With integrity and authentication, we can guarantee the correctness of the data collected.

Managing access to data is a crucial element of safeguarding information, as it specifies the individuals or entities permitted to view and manipulate transmitted data (solely the administrator is granted access in this scenario). The IoT base station recognizes that the data comes from the cluster head IoT device. But the IoT base station does not accurately identify the sender. This technique is known as anonymization, which protects the privacy of IoT devices.

The third phase of the SE-DTF architecture, namely HMAC-SHA1 signature and two-tier cryptography using ciphertext shifting with token-based access control, is proposed to achieve all the above requirements. When an IoT device (cluster member) senses anything, it creates data and encrypts it using Tier-1 cryptography. In Tier-1 cryptography, the packet is encrypted using a secret key based on the AES algorithm. This encoding provides ciphertext-1. Later this ciphertext was shifted.

Afterward, a cryptographic signature using HMAC-SHA1 is produced for the altered ciphertext utilizing the access control token provided by the IoT base station. This token allows authorized access to the data, while the signature ensures data integrity. The resultant packet combines the transformed ciphertext with the signature and is dispatched to the cluster head. The cluster head accumulates all incoming packets until the number of received packets reaches the packet aggregation threshold. Once the threshold is met, the cluster head forwards the aggregated packets to the next device along the MPCR path, which could be either a cluster head or a base station. The previous steps are repeated if the subsequent device is a cluster head.

But the next device is the base station; it transmits the aggregated packets to the base station. After receiving the aggregated packet, the base station encrypts it using Tier-2 cryptography. In Tier-2 cryptography, the aggregated packets are encrypted using a public key based on the RSA algorithm. It provides ciphertext-2. This ciphertext-2 is shifted once again. The base station then sends the shifted ciphertext to the administrator. After receiving the shifted ciphertext, the administrator recovers the ciphertext.

Subsequently, the recovered ciphertext is decrypted using the private key based on the RSA algorithm. It provides aggregated packets. Then the administrator splits these aggregated packets. After decryption, each of the shifted ciphertexts is recovered with HMAC-SHA1 signatures. If the administrator enters a valid token of each sender's IoT device, he can access each of the shifted ciphertexts; otherwise not (access control).

Additionally, the administrator creates new HMAC-SHA1 signatures for the shifted ciphertext using the entered token. The shifted ciphertext is secure if the new HMAC-SHA1 signature is equal to the received HMAC-SHA1 signature; otherwise, it is modified. After access control and signature verification, each ciphertext is recovered from each shifted ciphertext. Finally, each ciphertext is decrypted using the sender IoT device's secret key based on the AES algorithm. This encryption returns the original sensed data.

4 Results and Discussions:

The experimental findings and analysis of the secure and energy-efficient data transmission

framework (SE-DTF) in the IoT are presented in this part. Networks that were produced randomly are utilized for experimental studies. This simulation has as its starting point the uniform and random distribution of 100 IoT devices throughout a 900 m 600 m unit area, as shown in Figure 2. Each IoT device has a radio propagation range of 100 meters, and its initial energy is set at 100 J. A data payload capacity of 512 bytes has been designated. MATLAB is used to assess the SE-DTF framework.

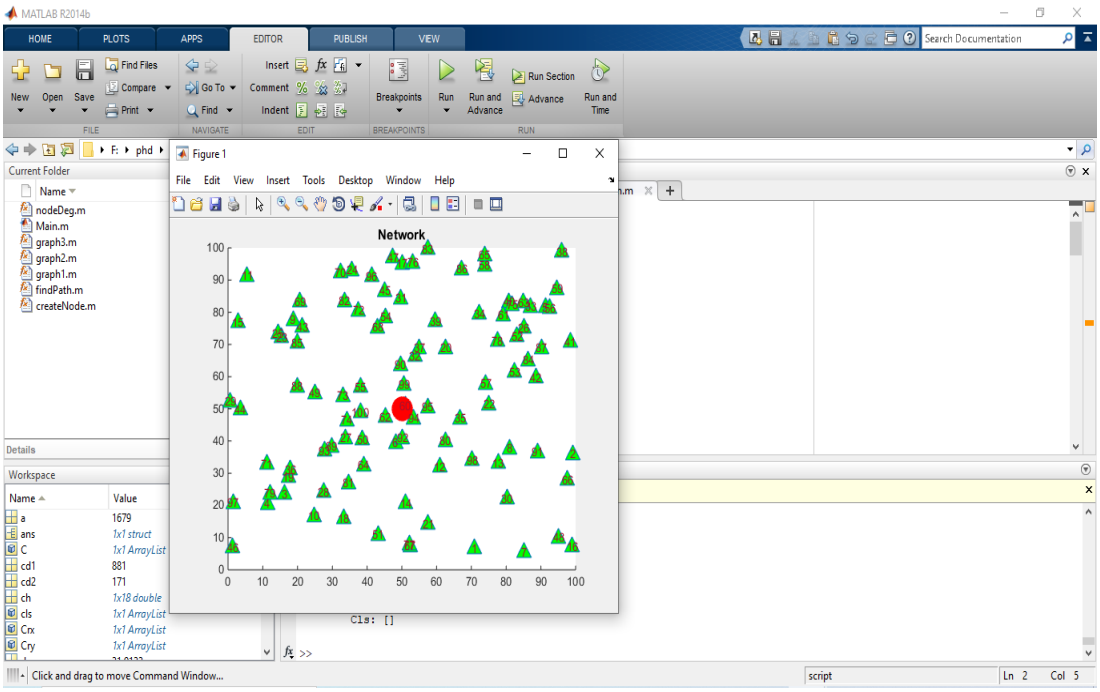

Figure 2: IoT network formation

4.1 Phase 1: IoT-PSKTS algorithm:

A comparison is conducted between this algorithm and other secret-sharing techniques to evaluate the performance of the proposed IoT-PSKTS algorithm. Specifically, AdiShamir's Perfect Secret Sharing Scheme (PSS), Hugo Krawczyk's Computational Secret Sharing scheme (CSS), and Rabin's Information Dispersal Algorithm (IDA) [16] are considered, and their share creation and recreation times are compared with those of the IoT-PSKTS algorithm. To assess the time required for share formation, Table 1 presents the time taken (measured in milliseconds) for generating shares, with n and k set at 5 and 3, respectively.

Table 1: The time (in milliseconds) required for generating shares (where n equals 5, and k equals 3).

Algorithm	Data size (in KB)		
	16	32	64
CSS	19.45	25.03	31.80
IDA	12.82	19.59	21.19
PSS	28.78	40.01	42.89
IoT-PSKTS	10.97	17.19	19.68

Figure 3 depicts a time-versus-data-sizes graph for creating shares when n = 5 and k = 3.

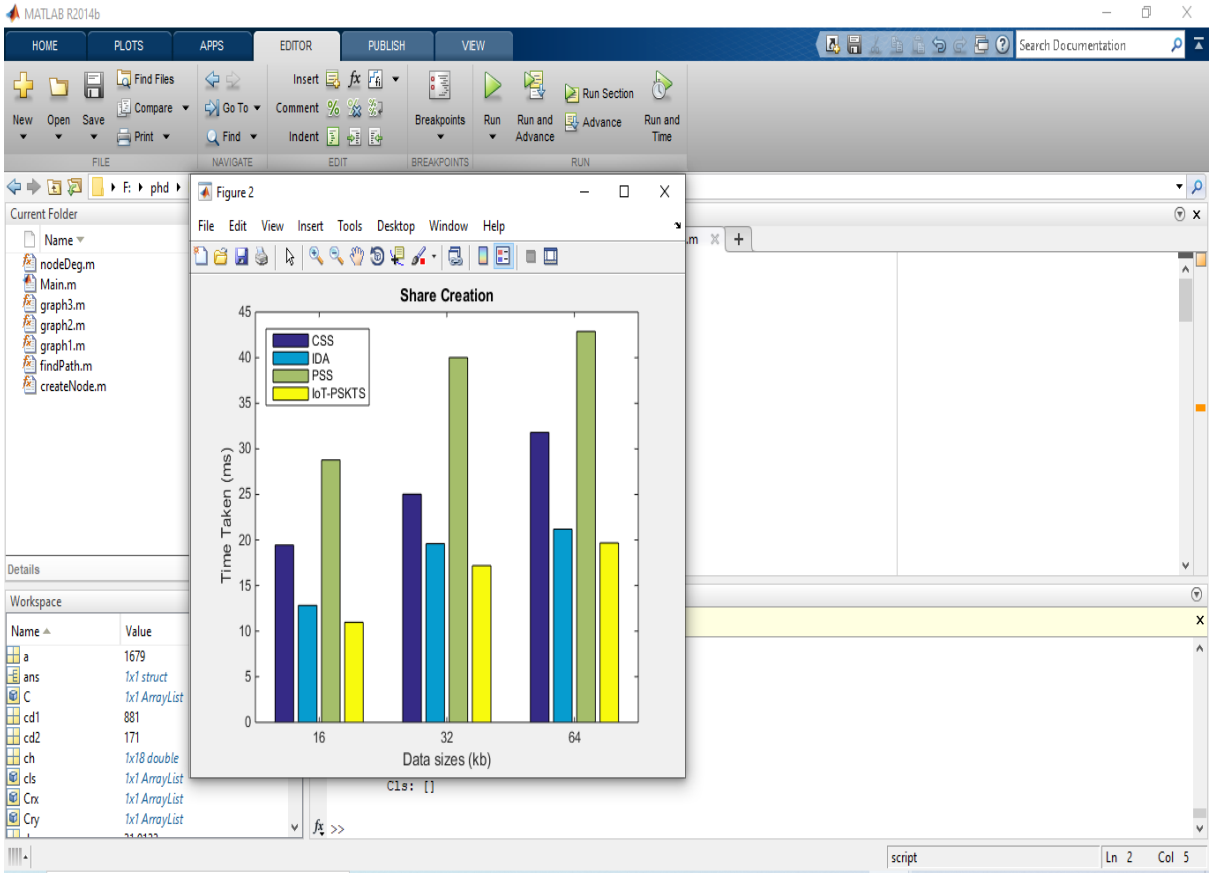

Figure 3: Visual representation indicating the relationship between the share creation time and data size is presented in the graph, where n is set to 5 and k is set to 3

Table 1 and Figure 3 illustrate that the IoT-PSKTS algorithm boasts the fastest performance among all algorithms, regardless of the size of the data being processed. Regarding how long it takes to create shares, IDA comes in second, CSS in third, and PSS last. One important finding from the results is that, compared to the other three algorithms, PSS exhibits more scalability problems as the size of the data increases. The duration required (in milliseconds; ms) for recreating shares (where n equals 5 and k equals 3) is exhibited in Table 2.

Table 2: The time duration (measured in milliseconds) for recreating shares with n set at 5 and k set at 3

Algorithm	Data size (in KB)		
	16	32	64
CSS	19.27	21.63	26.11
IDA	17.82	19.51	22.57
PSS	30.01	20.00	23.89
IoT-PSKTS	15.76	17.04	20.49

Figure 4 presents a graph illustrating the relationship between data size and time consumption for share recreation, with n and k assigned values of 5 and 3, respectively.

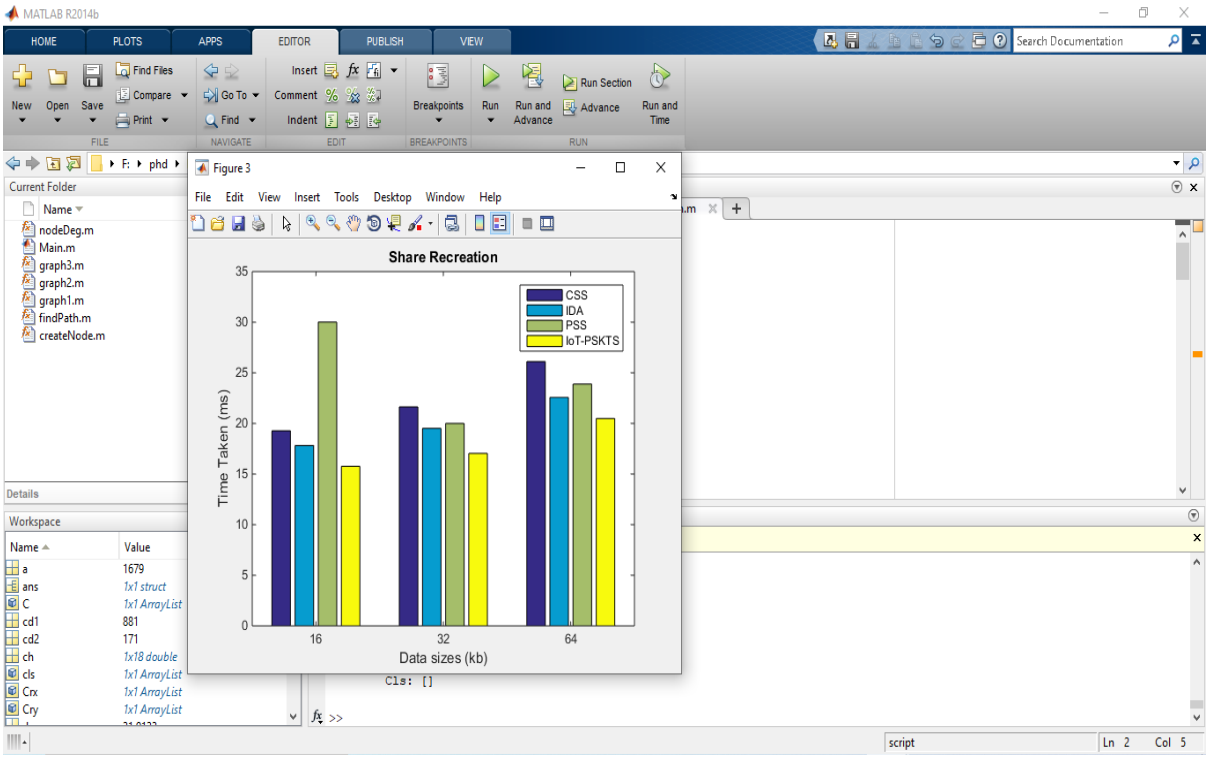

Figure 4: The relationship between data size and time consumption for share recreation, with n and k assigned values of 5 and 3

IoT-PSKTS is the quickest algorithm regardless of data size, as shown in Table 2 and Figure 4. However, regarding the time spent on share recreation, IDA comes in second, followed by CSS and PSS. Some noteworthy findings from the results include that, compared to the other three algorithms, PSS exhibits more scalability problems as the size of the data increases.

4.2 Phase 2: MPCR with the HFLC algorithm:

Furthermore, compare the MPCR with the HFLC algorithm with other routing approaches, such as InFRA [17], DRINA [17], and CBPR [17], in terms of throughput, packet delivery ratio, and energy utilization to assess the effectiveness of the MPCR with the HFLC algorithm. Throughput is the total number of data packets an algorithm might effectively send to the base station in a certain period. The throughput difference for MPCR using the HFLC algorithm, CBPR, DRINA, and InFRA at varying network device densities is demonstrated in Table 3 for each algorithm.

Table 3: Throughput Comparison

Number of IoT devices	InFRA	DRINA	CBPR	MPCR with HFLC
50	72	81	93	104
100	74	86	108	113
150	79	96	115	121
200	88	112	132	139

Figure 5 shows that the MPCR with the HFLC algorithm outperforms the InFRA, DRINA, and CBPR in terms of throughput for all network densities of device situations. Since the MPCR with the HFLC algorithm frequently uses just the optimal pathways while routing packets toward

the base station. The MPCR employing the HFLC algorithm chose the path with the highest residual power, the lowest power consumption rate, and the closest proximity to the base station. As a result, it demonstrates connection solidity and lowers packet drop rates when transmitting data. The proposed MPCR with the HFLC algorithm also uses the Fuzzy Logic cluster head selection technique to dynamically identify the best cluster head for use in transmitting aggregated packets. When MPCR with the HFLC algorithm considers everything above, it outperforms its rivals regarding throughput effectiveness.

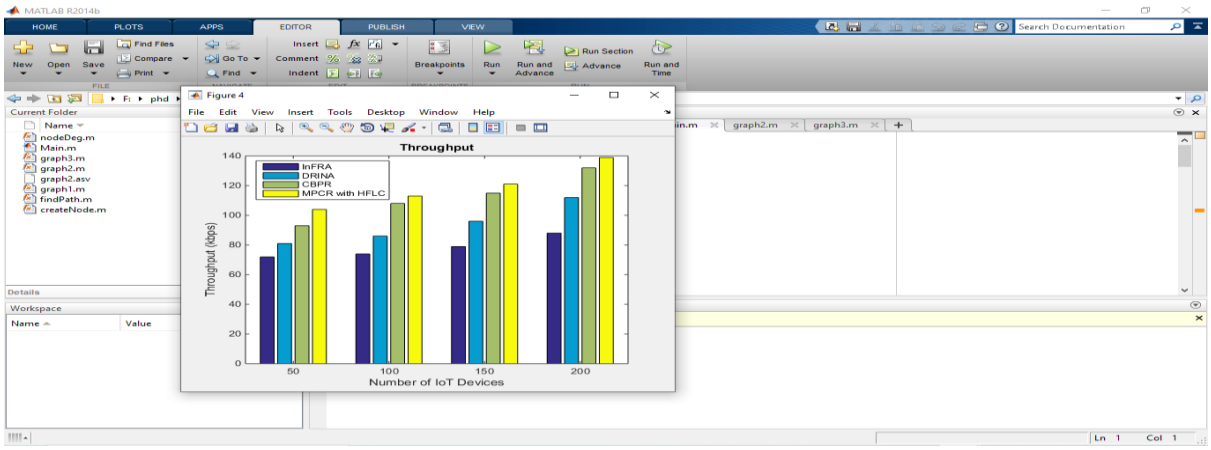

Figure 5: Throughput comparison

The packet delivery ratio is the ratio of packets obtained at the base station during the simulation period to the total number of packets transmitted from the source IoT device. Optimal integration and effective routing are indicated by high PDR. There are multiple route options between the source and destination devices, like the number of network devices and intermediary devices in the route.

By choosing intermediate cluster heads with more energy remaining to create the ideal forwarding routes with fewer chances of a link breakdown in the system, the MPCR with HFLC algorithm takes advantage of this multi-path factor to boost efficiency, guiding to competent transfer and a higher packet delivery ratio. Furthermore, to reduce congestion in the IoT network and shorten wait times, the MPCR with the HFLC algorithm prioritizes the packets based on the IoT device queue size condition, providing preference to the devices with a shorter queue size. As a result, it increases the PDR score. Additionally, unlike InFRA, DRINA, and CBPR routings, MPCR with the HFLC algorithm effectively utilizes data aggregation, which removes excessive data by integrating all associated event data into a solitary data unit. Table 4 compares packet delivery ratios.

Table 4: Comparison of Packet Delivery Ratios

Number of IoT devices	InFRA	DRINA	CBPR	MPCR with HFLC
50	38	46	64	76
100	39	49	66	79
150	40	60	72	81
200	44	63	76	84

Figure 6 demonstrates the PDR performances of the assessed systems, showing that each scheme's value increases as the number of devices increases.

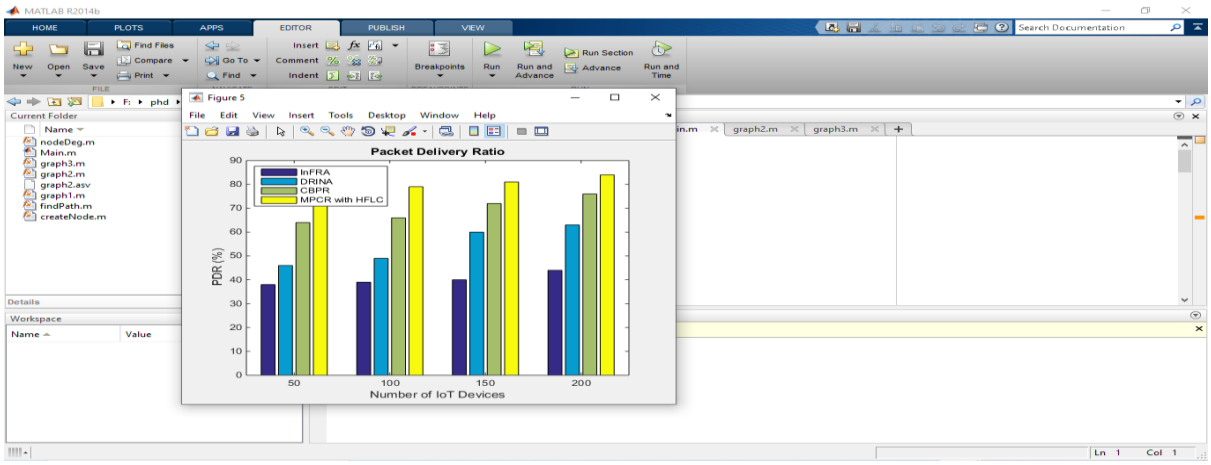

Figure 6: Packet delivery ratio comparison

Energy expenditure in an IoT system denotes the amount of power used to send a unit of information from a source IoT device to a base station. The efficiency of the MPCR with the HFLC algorithm's power utilization has been evaluated utilizing different network device densities and compared to the InFRA, DRINA, and CBPR routing systems, as demonstrated in Table 5 and Figure 7.

Table 5: Energy Consumption Comparison

Number of IoT devices	InFRA	DRINA	CBPR	MPCR with HFLC
50	0.59	0.38	0.23	0.19
100	0.46	0.34	0.17	0.15
150	0.43	0.31	0.14	0.12
200	0.37	0.29	0.13	0.1

MPCR, with the HFLC algorithm, always selects the shortest path to minimize energy consumption. Also, the better cluster head with higher residual energy is chosen closer to the base station and closer to all cluster members. Hence, it leads to a reduction in energy consumption. Subsequently, it used the packet aggregation technique. Hence, it leads to a reduction in unnecessary energy consumption.

Additionally, this algorithm also has a backup path. In case of connection failure, this algorithm automatically selects another shortest route. A lot of energy is saved this way. Therefore, compared to other proposed algorithms, MPCR with the HFLC algorithm effectively reduces energy consumption.

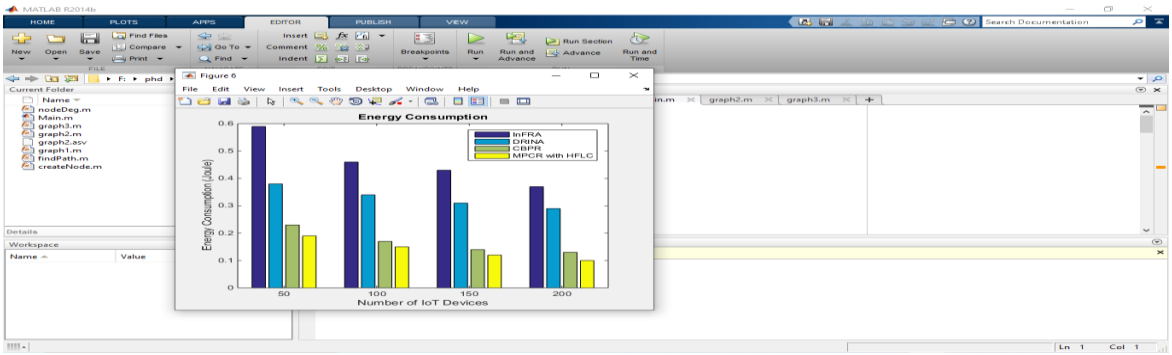

Figure 7: Energy expenditure comparison

4.3 Phase 3: Two-Tier Cryptography Technique:

To evaluate two-tier cryptography techniques, compare the proposed two-tier cryptography technique with existing cryptography techniques such as Blowfish [18] and AKCSS [18] regarding energy consumption and cryptography time. Table 6 depicts the energy expenditure comparison of the proposed two-tier cryptography technique with the existing cryptographic techniques.

Table 6: Energy Consumption Comparison

Technique	Energy Consumption (in microjoule/byes)
Blowfish	0.81
AKCSS	0.02692
Two Tier Cryptography	0.01571

Also, Figure 8 shows the comparison of energy consumption in graph form.

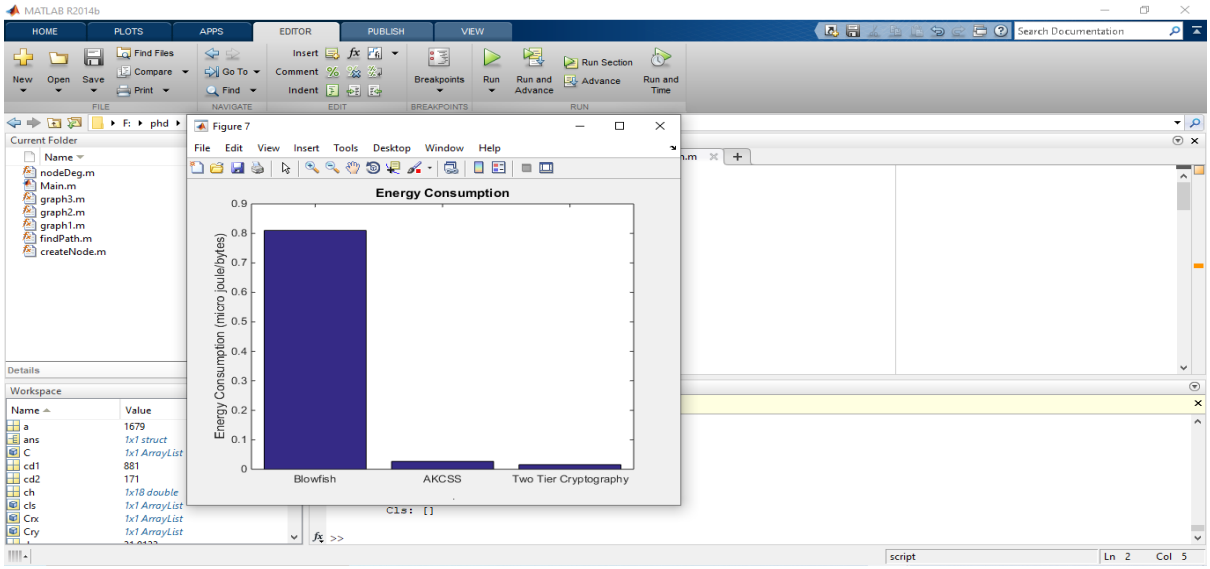

Figure 8: Energy Consumption Comparison

Figure 8 shows that the AKCSS technique consumes less energy compared to Blowfish. But compared to AKCSS, the proposed two-tier cryptography technique consumes much less energy for encryption. Since it uses lightweight cryptography (Tier-1 cryptography) in IoT devices. Also, Table 7 shows the encryption time comparison of the proposed two-tier cryptography technique with the existing cryptography techniques.

Table 7: Encryption Time comparison (in microseconds)

Message Size (in bits)	Blowfish	AKCSS	Two Tier Cryptography
100	9	2	1
500	23	12	9
1000	42	33	28
2000	88	82	76

Also, Figure 9 demonstrates the encryption time comparison in graph form.

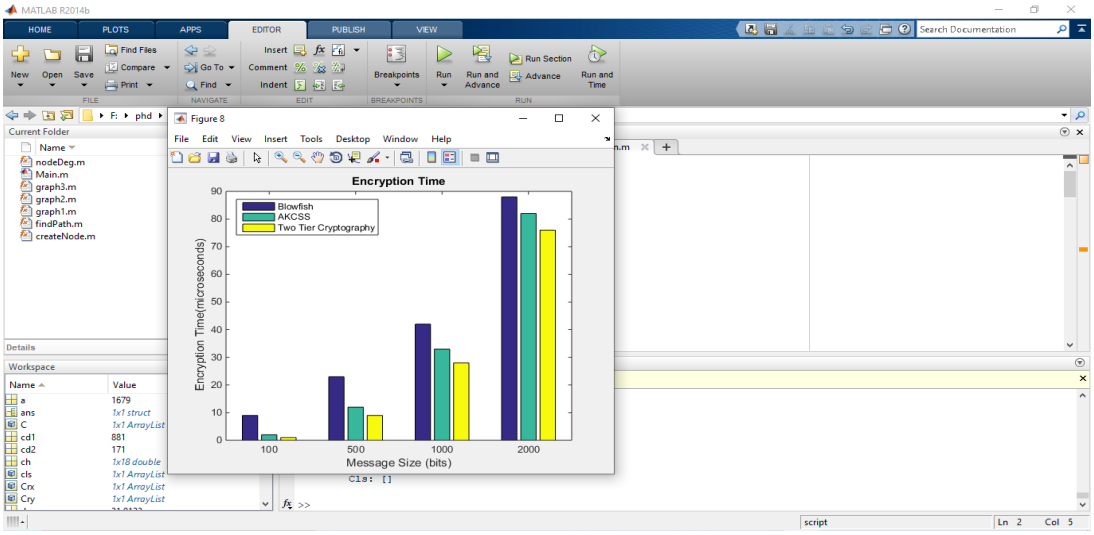

Figure 9: Comparison of encryption time

Figure 9 shows that compared to Blowfish, the AKCSS technique takes less time for encryption. But compared to AKCSS, the proposed two-tier cryptography technique takes less time for encryption. Because it splits the entire encryption process into two tiers and two devices to avoid the computational burden. The first tier is implemented at the sender IoT device and another at the IoT base station. Hence, it takes much less time than other encryption techniques. Also, Table 8 shows the decryption time comparison of the proposed two-tier and existing cryptography techniques.

Table 8: Decryption Time Comparison of Different Cryptography Techniques (in microseconds)

Message Size (in bits)	Blowfish	AKCSS	Two Tier Cryptography
100	4	2	1
500	20	16	12
1000	38	23	20
2000	82	53	48

Also, Figure 10 shows the decryption time comparison in graph form.

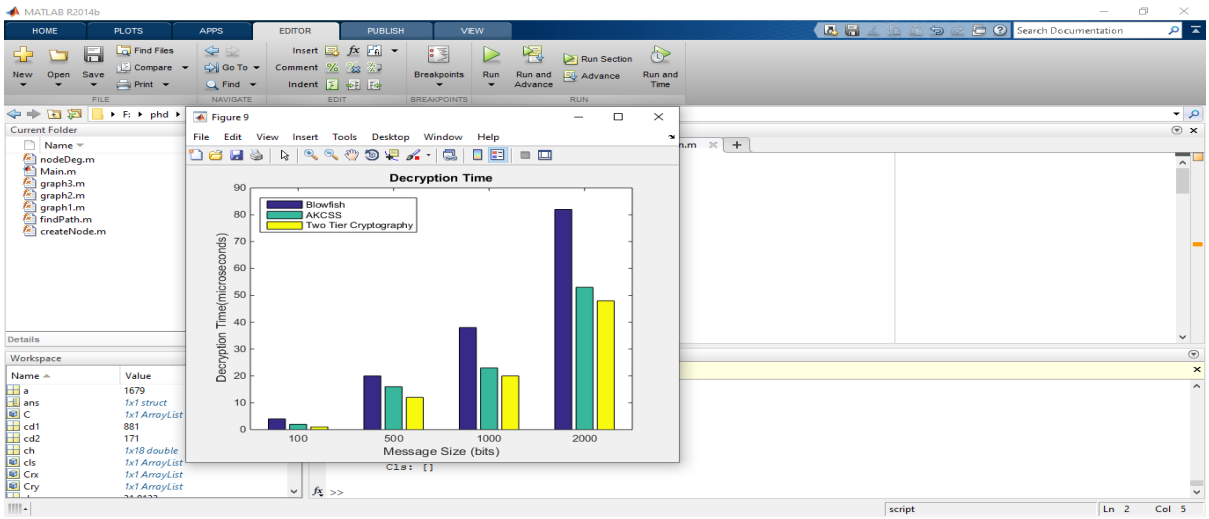

Figure 10: Decryption Time Comparison

Figure 10 shows that compared to Blowfish, the AKCSS technique takes less time for decryption. But compared to AKCSS, the proposed two-tier cryptography technique takes much less time for decryption.

5 Conclusion:

Key exchange is a method in cryptography that allows the usage of a cryptographic technique by facilitating the swap of cryptographic keys among two entities. Both the IoT devices and the admin should be able to message encryption and decryption if they are to swap encrypted messages. The kind of tools they require vary depending on the encryption technique they should utilize. Both will require a duplicate of a similar key if they utilize one. The key swap issue outlines how to swap any keys or other data required for creating a safe communication channel to prevent others from getting a copy. The sensed data is captured, encrypted, and delivered to the base station, where it is aggregated before being forwarded to the administrator for additional processing. The data is transmitted to the base station using various routing techniques with less power. When using low-power IoT devices to develop a routing algorithm, energy efficiency should also is considered a critical performance metric. Consequently, this paper presented a secure and energy-efficient data transmission framework (SE-DTF) for the IoT. There are 3 phases to this framework. The public and secret key with token sharing (IoT-PSKTS) algorithm is employed in the initial phase to stop key leakages in the IoT. The second phase combines the Minimum Power Usage Routing (MPCR) algorithm and the Hierarchical Fuzzy Logic Clustering (HFLC) algorithm to reduce power consumption. Safe data transfer is the main goal of the third phase, which employs two-tier cryptography, ciphertext shifting, token-based access control, and HMAC-SHA1 signature. The experimental results showed how safely the IoT-PSKTS algorithm might share a token with a public and a secret key. The MPCR using the HFLC algorithm was also shown to surpass other current algorithms regarding throughput, packet delivery ratio, and energy expenditure. Additionally, it showed that compared to other existing cryptography techniques, the two-tier cryptography technique utilized less energy and required less computation time for encryption and decryption.

References:

[1] Sankar, S., & Srinivasan, P. (2018). Multi-layer cluster-based energy-aware routing protocol for the IoT. Cybern. Inf. Technol, 18(3), 75-92.

[2] Sujanthi, S., & Kalyani, S. N. (2020). SecDL: QoS-aware secure deep learning approach for dynamic cluster-based routing in WSN-assisted IoT. Wireless Personal Communications, 114(3), 2135-2169.

[3] Aranzazu-Suescun, C., & Cardei, M. (2019). Anchor-based routing protocol with dynamic clustering for IoT WSNs. EURASIP Journal on Wireless Communications and Networking, 2019(1), 1-12.

[4] Maheswar, R., Jayarajan, P., Sampathkumar, A., Kanagachidambaresan, G. R., Hindia, M. N., Tilwari, V., ... & Amiri, I. S. (2021). CBPR: A cluster-based backpressure routing for the IoT. Wireless Personal Communications, 1-19.

[5] Li, J., Silva, B. N., Diyan, M., Cao, Z., & Han, K. (2018). A clustering-based routing algorithm in IoT-aware Wireless Mesh Networks. Sustainable cities and society, 40, 657-666.

[6] J. Tang, H. Song, A. Xu, Y. Jiang, H. Wen, Y. Zhang, K. Qin, "Secret Sharing Simultaneously on the Internet of Things," 2020 IEEE International Conference on Power, Intelligent Computing and Systems, ISBN: 978-1-7281-9874-3, Jul 2020.

[7] M. Farhadi, H. Bypour, R. Mortazavi, "An efficient secret sharing-based storage system for cloud-based IoT," 2019 IEEE 16th International ISC Conference on Information Security and Cryptology, ISBN: 978-1-7281-4374-3, Aug 2019.

[8] Maheswar, R., Jayarajan, P., Sampathkumar, A., Kanagachidambaresan, G. R., Hindia, M. N., Tilwari, V., ... & Amiri, I. S. (2021). CBPR: A cluster-based backpressure routing for the IoT. Wireless Personal Communications, 1-19.

[9] Kumar, N. P. R., & Gnanadhas, J. B. (2020). Cluster Centroid-Based Energy Efficient Routing Protocol for WSN-Assisted IoT.

Advances in Science, Technology, and Engineering Systems Journal, 5(4), 296-313.

[10] Sankar, S., & Srinivasan, P. (2018). Multi-layer cluster-based energy-aware routing protocol for the IoT. Cybern. Inf. Technol, 18(3), 75-92.

[11] Zhang, K., Long, J., Wang, X., Dai, H. N., Liang, K., & Imran, M. (2020). Lightweight searchable encryption protocol for the industrial IoT. IEEE Transactions on Industrial Informatics, 17(6), 4248-4259.

[12] A. Yousefi, S. M. Jameii, "Improving the security of IoT using encryption algorithm," 2017 IEEE International Conference on IoT and Application, ISBN: 978-1-5386-1698-7, May 2017.

[13] Choudhary, K., Gaba, G. S., Butun, I., & Kumar, P. (2020). Make-it—a lightweight mutual authentication and key exchange protocol for the industrial IoT. Sensors, 20(18), 5166.

[14] Schuster, R., Shmatikov, V., & Tromer, E. (2018, October). Situational access control in the IoT. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 1056-1073).

[15] A. Terkawi, N. Innab, S. A. Amri, A. A. Amri, "IoT Increasing the Necessity to Adopt Specific Type of Access Control Technique," 2018 IEEE 21st Saudi Computer Society National Computer Conference, ISBN: 978-1-5386-4110-1, Apr 2018.

[16] W. J. Buchanan, D. Lanc, E. Ukwandu, L. Fan, G. Russell, and O. Lo, "The Future Internet: A World of Secret Share," Future Internet 2015, 7, 445-464, DOI:10.3390/fi7040445.

[17] Maheswar, R., Jayarajan, P., Sampathkumar, A., Kanagachidambaresan, G. R., Hindia, M. N., Tilwari, V., ... & Amiri, I. S. (2021). CBPR: A cluster-based backpressure routing for the IoT. Wireless Personal Communications, 1-19.

[18] Preethi, R., & Sughasiny, M. (2018, December). AKCSS: An Asymmetric Key Cryptography Based on Secret Sharing in Mobile Ad Hoc Network. In International Conference on Intelligent Systems Design and Applications (pp. 73-86). Springer, Cham.