Reducing Cybersecurity Risks in Cloud Computing Using A Distributed Key Mechanism

Saleh M. Altowaijri†
saleh.altowaijri@nbu.edu.sa
Department of Information Systems, Faculty of Computing and Information Technology
Northern Border University, Rafha, Kingdom of Saudi Arabia

Summary
The Internet of things (IoT) is the main advancement in data processing and communication technologies. In IoT, intelligent devices play an exciting role in wireless communication. Although, sensor nodes are low-cost devices for communication and data gathering. However, sensor nodes are more vulnerable to different security threats because these nodes have continuous access to the internet. Therefore, the multiparty security credential-based key generation mechanism provides effective security against several attacks. The key generation-based methods are implemented at sensor nodes, edge nodes, and also at server nodes for secure communication. The main challenging issue in a collaborative key generation scheme is the extensive multiplication. When the number of parties increased the multiplications are more complex. Thus, the computational cost of batch key and multiparty key-based schemes is high. This paper presents a Secure Multipart Key Distribution scheme (SMKD) that provides secure communication among the nodes by generating a multiparty secure key for communication. In this paper, we provide node authentication and session key generation mechanism among mobile nodes, head nodes, and trusted servers. We analyzed the achievements of the SMKD scheme against SPPDA, PPDAS, and PFDA schemes. Thus, the simulation environment is established by employing an NS 2. Simulation results prove that the performance of SMKD is better in terms of communication cost, computational cost, and energy consumption.

Keywords:
IoT, Cloud Computing, Multiparty Key, Secure Communication, Key Establishment, Security.

1. Introduction

In IoT-enable wireless sensor networks, intelligent devices play an essential role in secure communication and data transmission [1]. However, IoT gains a lot of attention by enhancing the support of intelligent devices to provide more efficient and reliable solutions for different fields [2]. Although, it also opens a path for several challenging issues. IoT application scenario is shown in Figure 1. Mostly, intelligent devices have continuous access to the internet. Therefore, the continuous connectivity of sensor nodes with the internet makes them more vulnerable to different security threats. Thus, an efficient and reliable solution is required for secure data exchange [3]. In this context, a group-based approach is considered for secure communication among intelligent devices [4]. A group-based approach is considered in several fog-assisted and cloud-based schemes. Moreover, a secure key-based encryption and decryption mechanism is mandatory among the nodes for secure communication [5].

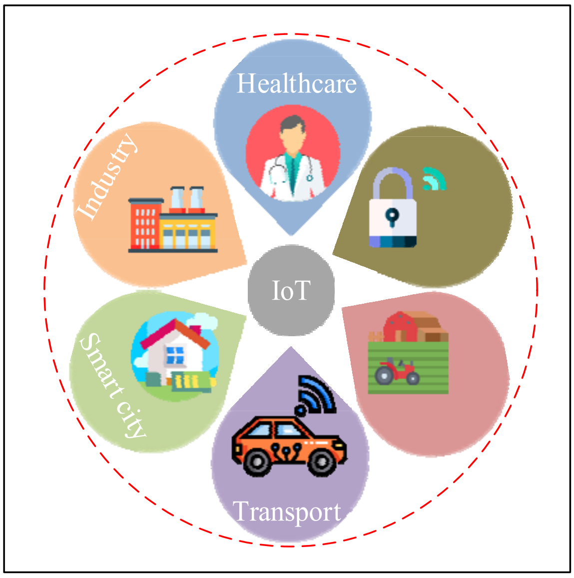

Fig 1 IoT Applications

A multiparty key generation mechanism provides secure communication and data privacy because it considers the security credentials of group members for a key generation[6]. Multiparty key generation mechanism is a challenging task because the key generation mechanism is dependent on neighbor intelligent devices [7]. In this context, intelligent devices can communicate with other devices by utilizing an open channel. The attackers can also gain passage through open channels to access the data. Thus, an attacker can perform different attacks like DOS attack, reply attack, and Man-in-the-middle attack by using an open channel [8]. In this situation, a reliable and secure key generation mechanism is required that shielding against different security attacks.

Cybersecurity is essentially important because intelligent devices are mostly deployed in a hostile

Manuscript received September 5, 2021
Manuscript revised September 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.9.1

environment. Thus, a security scheme is required that provides node authentication, data confidentiality by employing symmetric and asymmetric key-based encryption [9]. Moreover, security and privacy-based schemes are employed message hashes to protect the message's integrity. The Chebyshev polynomials are used to provide the best estimation for continuous functions. For security and efficiency, it enhances the performance of authentication and session key generation [10]. Several schemes provide data validation with reduced computational cost, but these schemes do not properly consider the security for communication scenarios. The main problem in multipart key generation and authentication-based schemes is the extensive multiplications and calculations. RSA and ECC-based schemes required extensive calculations and these calculations are more complexed when the number of parties increased [11]. Moreover, several schemes utilize the Diffie-helmen-based Key generation procedure for authentication. Hence, there is a probability for the man-in-the-middle attack [12].

In this paper, we present a secure multiparty key distribution (SMKD) method for efficient authentication and secure key generation. Moreover, Secure Hash Algorithm (SHA) 2 is considered to protect the message while transmitting. Furthermore, the proposed work is divided into three phases. In phase-1, mobile nodes (Mn) share security credentials with head nodes (Hn) where Hn authenticate each Mn by checking the received security credentials and pre-forwarded random values. After that, a session key is generated among the mobile nodes and Hn without sharing the whole key over the network. In phase 2, the session key generation procedure is considered among the head nodes. In this context, a common session key is generated by exchanging security credentials. In phase 3, head nodes are authenticated at the trusted server (TS). Moreover, Hn shares security credentials and pre-received random values with the TS. Then, TS and Hn generate the same session key from received and forwarded security credentials. The proposed scheme simulation environment is implemented in NS.2. Moreover, SMKD is analyzed with other key generation-based methods and results illustrate the supremacy of the SMKD scheme.

We have organized the rest of the paper structure as. Section 2 explores the different multipart-based key generation mechanisms. Section 3 describes the framework multiparty key generation process. Section 4 describes the SMKD authentication and session key generation mechanisms. Section 5 illustrates the performance of the proposed scheme and analyzes it against counterparts. In the end, section 6 is the conclusion section of the paper.

2. Related Work

In open channels, security and privacy are essential elements for secure communication. Therefore, we consider different secret key generation-based schemes that involve security credentials for authentication and session key generation. Aness et al. discuss a bilinear Elgamal cryptosystem for a secure and privacy-preserved scheme. Diffie-Hellman-based assumptions are employed for secure communication [12]. The Multiparty key management scheme provides authentication and session key generation for communication over an open channel. A user can be used a symmetric key or an open key and also uses a combination of both for secure communication and data transmission [13]. An anonymity-based scheme is discussed to provide user anonymity. Moreover, Data encryption is used to shielding against different security attacks [14]. Xiong et al. [15] present a privacy preservation scheme that utilizes asymmetric key-based encryption and also provides anonymity and authentication to protect data integrity. P3-PAKE method acknowledges the selection hypothesis as a security credential. The enhanced communication and computational cost of P3-PAKE also affect the energy consumption of sensors nodes [16]. The M-PAKE scheme is a key establishment from a multiparty password-based authentication and the user password is stored at the server. In the case of user password establishment at trusted sever session keys are established for each user. Moreover, the trusted server holds the password of all users and is forwarded to the users securely [17].

PFDA [18] presents an enhanced symmetric key-based homomorphic cryptosystem. Fog based approach is applied for effective and secure monitoring of medical information. The Diffie-Hellman assumption is considered to resist against several security threats. It also handles emergency scenarios by providing effective methods. Xiong et al. present a key generation-based scheme for authentication and anonymity for WBANs. The sensor nodes are placed on the patient body to anonymously generating session keys and data forwarding. BAN logic and informal method are utilized for validation [19]. Hong et al. discussed a privacy preservation scheme for authentication of smart healthcare devices identity authentication. A Min hash-based authentication method is employed to preserve data integrity. The secure ciphertext is utilized for secure data transmission. GNY logic-based analysis is conducted for validation [20]. UAKMP provides a secure and lightweight user authentication at a remote location by generating session keys for communication. key generation mechanism used three security credentials for session key generation. AVISPA tool is employed for security validation. Moreover, it provides effective communication and computational cost during node anonymity and node registration [21]. Wazid et al. discuss a blockchain-assisted reliable key management method for the internet of intelligent things [22].

Xiaodong et al, [23] provide a fog-assisted and privacy preservation scheme for data aggregation and transmission. A secure aggregation method is considered at the cloud server that received all data while preserving the privacy of sensor nodes. Moreover, the fog node provides false data filtering and data aggregation for saving the bandwidth while forwarding the data towards the cloud server. At the fog node, the aggregate signatures are utilized for data authentication. It also protects the integrity of the data integrity in the case of dynamic groups. Gowithami et al. [24] discussed an effective three factors authentication scheme that employs XOR and a one-way hash function for authentication. Three factors are three security credentials that are used for authentication. Session keys are considered for secure communication among the nodes. Furthermore, it protects from different security attacks and the AVISPA tool is used for security validation.

Xiaoying et al. provide a bilinear parring-based three-party AKA method for authentication and session key generation. A random oracle method is employed for the validation of the security method. The security analysis proves that it shielded against different security attacks [25]. Hong et al. present a strong and effective information gathering method that utilizes symmetric key-based cipher to preserve data privacy and anonymity. An effective signature scheme is employed for authentication. The base station received encrypted data from end nodes. The data aggregation and data decryption methods are employed at the base station [26]. Omar et al. present a privacy preservation method that utilized a homomorphic encryption method to provide privacy. It also checks the integrity of the data at every hope and drops the data due to integrity violation. In this context, the verification method is based on the Tiny ECC method. The simulation is conducted in TelosB and MicaZ to prove the supremacy [27].

3. System Model

In this section, we present our proposed node authentication and secure communication model for group-based scenarios in different fields. A proposed framework that provides secure communication from mobile nodes to the trusted server is illustrated in Figure 1. In this model, each group has one Hn and several mobile nodes. Hn authenticates all the mobile nodes in the group by securely sharing the random numbers with the Hn. Then, Mn constructs an encrypted message that includes pre-received random numbers and forwards this message to the Hn. Mn authenticated based on the received random numbers. After authentication, Hn and Mn establish a session key for secure communication by using security credentials for key generation. The Hn_i also communicate with Hn_j, for secure communication security credentials are shared with each other in an encrypted message for multiparty session key generation. Moreover, The Hn are authenticated at the TS. For authentication, Hn securely received random numbers from TS, and Hn forwards these random numbers and other security credentials to the TS in an encrypted message. For secure communication, both nodes establish a secure key by using security credentials to generate a session key.

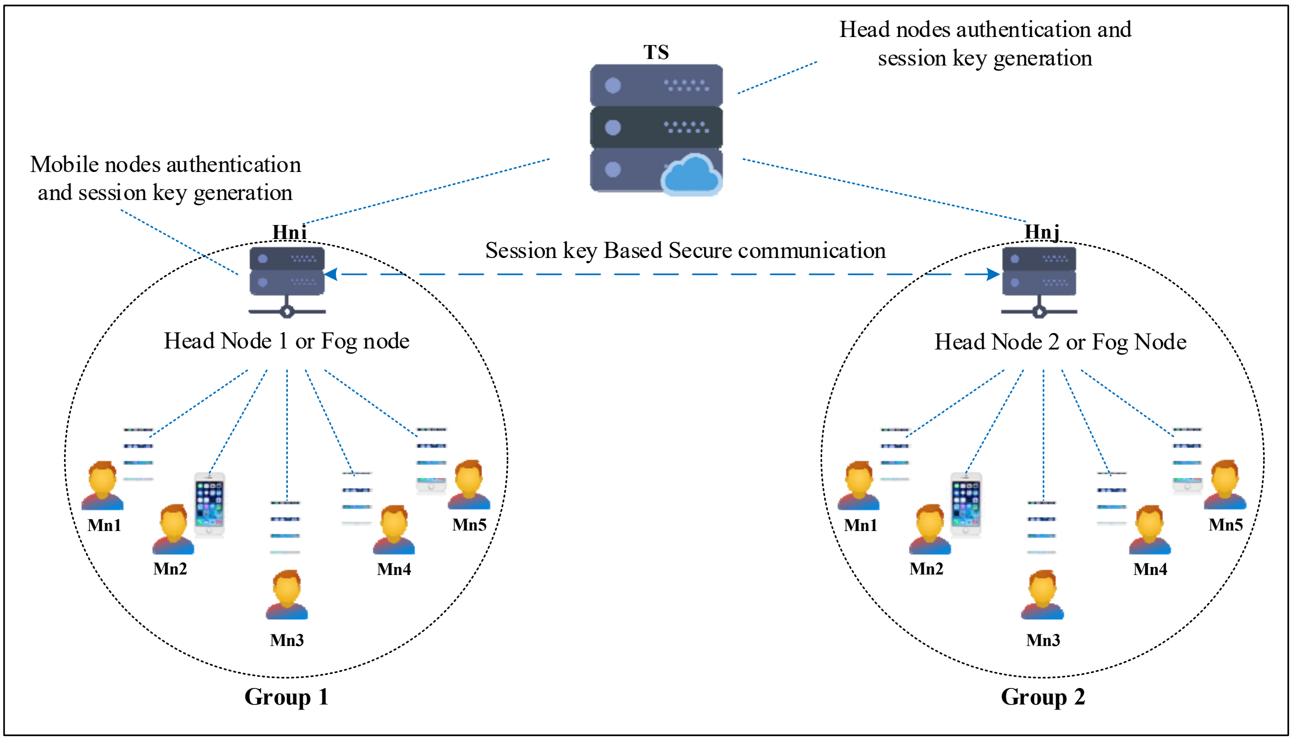

Fig 2 System Model

In our communication model, WiFi technology is adopted for communication among the nodes. In our communication model, we use the symmetric key-based encryption method for message encryption. All mobile nodes in the same group are communicating with a single Hn. The head nodes authenticate each node in the group. A single TS is used for the head nodes. authentication and Hn_i can communicate with the Hn_j of another group. The main problem in a multiparty authentication scheme is the extensive multiplication. When the number of parties increased the multiplications are more complex. Thus, the computational cost of batch key and multiparty key-based schemes is high. Therefore, an effective and secure multiparty key mechanism is required for secure communication.

4. Secure multiparty Key Distribution (SMKD) Scheme

The Secure Multipart Key Distribution scheme (SMKD) provides privacy preservation for IoT-based intelligent devices. SMKD is appropriate for the group-based situation in several fields where a group of intelligent nodes is registered and authenticated through a fog node. In our proposed system model, a fog node is acting as a Head node (Hn). Several (Mn) are registered at the Hn to form a group. For security, the TS generates a key among the sender node and receiver node by employing the security credentials of several parties. For confidentiality, secret keys are generated during message exchange. A trusted server is considered for authentication and multiparty session key generation for secure communication with a number of head nodes. Moreover, symbols that are used in the paper are listed in Table 1.

Table 1: List of Symbols

Symbol	Description
Mn_{id_i}	Mobile node Id
PW_{MN_i}	Mobile node password
Rn	Random Numbers
Sc_{Mn_i} ,Sc_{Hn_i}	Symmetric key at Mn_i and Hn_i
Ts	Time stamp
N_{Mn_i}, N_{Hn_i}	Nonce value at Mn_i and Hn_i
Em_{Mn_i}	Encrypted message at Mn_i
Rm	Authentication message of Mn_i
Ca_{Hn_i}	Secret Key at Hn_i
SM_{Hn_i}	Encrypted message of node Hn_i
AM_{Hn_i}	Authentication message of Hn_i
SK_{Hn-Mn}	Session key between Mn_i and Hn_i

Our proposed multipart key distribution method for secure communication is divided into 3 phases for secure communication. In phase 1, we provide authentication and session key generation among the head node (Hn) and mobile nodes (Mn). Mn nodes are registered to a single Hn and then Mn generates a message that contains the security credential of Mn. Moreover, a hash function is employed to securely forward these values to Hn. By employing these parameter values, Hn generates the same session key to communicate with the mobile nodes. In phase 2, our proposed scheme provides secure communication among the head nodes of two different groups. In phase 3, the head node is authenticated with a trusted server and also generates a multiparty session key for secure communication.

4.1 Mobile Node Authentication and Session Key Generation

In node authentication, Mn_i securely receives a message Sc_{Mn_i} (Rn { H(Rn)}) that contains random numbers (Rn) and H(Rn) from the Hn_i that are encrypted with Mn_i symmetric key (Sc_{Mn_i}). At each Mn_i the received hash (Rn)′ is compared with the calculated hash (Rn) of the Mn_i. Then, form an encrypted message $Em_{Mn_i} =$ Rn $\oplus$ Sc_{Hn_i} by using the XOR among Rn and (Sc_{Hn_i}) to form an Em_{Mn_i} as shown in algorithm 1.

Algorithm 1:*Mn* Node Authentication at *Hn* & key Generation

1. $Hn_i \rightarrow Mn_i$: Sc_{Mn_i} (Rn { H(Rn)}) **then**
2 Mn_i: **If** H(Rn)′ equals H(Rn) **then**
3. Mn_i: $Em_{Mn_i} =$ Rn $\oplus Sc_{Hn_i}$
4. $Mn_i \rightarrow Hn_i$: $Rm = Sc_{Hn_i}$ {(Mn_{id_i}, Ts_{Hn_i}, Em_{Mn_i}, PW_{MN_i}, N_{Mn_i}),$H(Rm)$} **after that**
5. **Else**
6. Drop the message due to integrity violation
7. **End if**
8. Hn_i: **If** (Ts_{Mn_i}- Ts_{Hn_i}) < Δt **then**
9. Hn_i: **If** $H(M_{Ni})$ equals $H(M_{NA})$ **then**
10. $Em_{Mn_i} \oplus Sc_{Hn_i}$ to extract Rn
11. Hn_i: **If** H(Rn)′ equals H(Rn) **then**
12. Hn_i: Mn_i authenticated successfully
13. Hn_i: RK_{Hn-Mn} = {Rn$_1$, Rn$_2$, ... Rn$_r$}
14. Hn_i: SK_{Hn-Mn} = H(Mn_{id_i}) $\oplus$ H(PW_{MN_i}) $\oplus$ H(RK_{Hn-Mn}) $\oplus$ H(Rn)
15. **Else**
16. Discard message
17. **End if**
18. **Else**
19. Drop the message due data falsification
20. **End if**
21. **Else**
22. Discard message due to freshness failure
23. **End if**

Then Mn_i forwards its security credentials to the Hn_i through an authentication message as given in equation 1.

$$Rm = Sc_{Hn_i}\{(Mn_{id_i}, Ts_{Hn_i}, Em_{Mn_i}, PW_{MN_i}, N_{Mn_i}), H(Rm)\} \quad (1)$$

The authentication message Rm contains mobile node id (Mn_{id_i}), the hash of random numbers (Em_{Mn_i}) encrypted random numbers, password of a mobile node (PW_{MN_i}) and nonce value (N_{Mn_i}).

The Hn_i checks the timestamp by calculating the difference between forwarded and received timestamps. It ignores the message when the difference is high. Otherwise, the Hn_i checks the message hashes by matching the hash $H(Rm)'$ of forwarded parameters with the calculated hash $H(Rm)$ of the acquired parameters at Hn_i. After that, Em_{Mn_i} is $\oplus$ with Sc_{Hn_i} to extract the Rn from the message and compare this received Rn$'$ with the Rn that is initially forwarded to the Mn_i. Moreover, In the case of Rn matches, the Hn_i verifies the successful authentication with Mn_i otherwise, discard the received message.

After successful authentication of all mobile nodes, Hn_i select (Rn) values$\{\mathrm{Rn}_1, \mathrm{Rn}_2, \ldots \mathrm{Rn}_r\}$ from a random number of mobile nodes to form a random key RK_{Hn-Mn} and use other security credentials like ($Mn_{id_i}, PW_{MN_i}, N_{Mn_i}, Rn$) of the Mn to form a session key among the Hn_i and a group of Mn_i as shown in equation 2.

$$SK_{Hn-Mn} = \mathrm{H}(Mn_{id_i}) \oplus \mathrm{H}(PW_{MN_i}) \oplus \mathrm{H}(RK_{Hn-Mn}) \oplus \mathrm{H}(Rn) \quad (2)$$

Furthermore, Hn_i share the RK_{Hn-Mn} with the group of mobile and mobile nodes using RK_{Hn-Mn} and other security credentials to construct the same session key without sharing the whole key over the open channel.

4.2 Session Key Generation Between Head Nodes

In session key generation between two head nodes, both head nodes share a hash of the secret key Ca_{Hn_i} by employing symmetric key-based encryption. In algorithm 2, encrypted message of Hn_i forwards to the Hn_j as given in equation 3.

$$SM_{Hn_i} = Sc_{Hn_j}\{(Hn_{id_i}, Ts_{Hn_i}, N_{Hn_i}, H(Rn_i), H(Ca_{Hn_i}), \mathrm{H}(SM_{Hn_i})\} \quad (3)$$

Moreover, Hn_j also sends the SM_{Hn_j} message towards the Hn_i as shown in equation 4.

$$SM_{Hn_j} = Sc_{Hn_i}\{(Hn_{id_j}, Ts_{Hn_j}, N_{Hn_j}, H(Rn_j), H(Ca_{Hn_j}), \mathrm{H}(SM_{Hn_j})\} \quad (4)$$

The encrypted messages contain the id of the head node (Hn_{id}), timestamp (Ts_{Hn}), nonce value (N_{Hn}), $H(Rn_i)$ random numbers, and the hash of the encrypted message $H(SM_{Hn})$. At Hn_j, the received message is decrypted by employing a symmetric key Sc_{Hn_i}. Furthermore, Hn_j is checking the timestamp of the received message SM_{Hn_j} to check message freshness and also comparing the hashes of the received message to checking the integrity of the received message.

After that, Hn_j create a session key by taking the hash of the shared secret key H(Ca_{Hn_i}), the hash of head nodes id H(Hn_{id_i}) H(Hn_{id_j}), and the hash of random numbers H(Rn) and also taking the XOR of these values to create a session key as given in equation 5.

$$SK_{Hn_i-Hn_j} = H(Hn_{id_i}) \oplus H(Hn_{id_j}) \oplus H(Ca_{Hn_i}) \oplus H(Rn) \quad (5)$$

Furthermore, Hn_i also conducting the same steps for constructing the session key. In this context, both Hn_i and Hn_j are constructing the same session key for communication.

Algorithm 2: Session key Generation Between Hn_i & Hn_j

---

1. $Hn_i \rightarrow Hn_j$: $SM_{Hn_i} = Sc_{Hn_j}\{(Hn_{id_i}, Ts_{Hn_i}, N_{Hn_i}, H(Rn_i), H(Ca_{Hn_i}), \mathrm{H}(SM_{Hn_i})\}$ **after that**
2. Hn_j: **If** (Ts_{Hn_i} - Ts_{Hn_j}) < Δt **then**
3. Hn_j: **If** $H(M_{Ni})$ equals $H(M_{NA})$ **then**
4. Create session key by taking XOR of security credentials
5. Hn_j: $SK_{Hn_i-Hn_j} = \mathrm{H}(Hn_{id_i}) \oplus \mathrm{H}(Hn_{id_j}) \oplus \mathrm{H}(Ca_{Hn_i}) \oplus \mathrm{H}(Rn)$
6. **Else**
7. Drop the message due data falsification
8. **End if**
9. **Else**
10. Discard message due to freshness failure

4.3 Head Nodes Authentication and Session Key Generation

Initially, Hn_i securely receives random numbers (Rn) and H(Rn) from TS. Then Hn_i comparing the hash of received Rn$'$ with calculated Rn. In case hashes are matched, then Em_{Hn_i} = Rn $\oplus$ Sc_{TS} by taking an $\oplus$ of Rn with a symmetric key (Sc_{TS}) to form an Em_{Hn_i}. Then, Hn_i construct an authenticated message as given in equation 6.

$$AM_{Hn_i} = Sc_{TS}\{Am = [(Hn_{id_i}, Ts_{Hn_i}, N_{Hn_i}, Em_{Hn_i})], \mathrm{H}(Am)\} \quad (6)$$

The authentication message contains the id of Hn_i (Hn_{id_i}), timestamp (Ts_{Hn_i}), nonce value (N_{Hn_i}), (Em_{Hn_i}) encrypted random numbers and the hash of the authentication message $H(Am)$. Then, Hn_i encrypts the message by using symmetric Sc_{TS} and forwards the encrypted message to the trusted server (TS). On the other hand, TS checking the timestamp of the received message

Cm_{Hni} to check message freshness and also comparing the hashes to checking the message integrity. After that, Em_{Hn_i} is $\oplus$ with Sc_{TS} to extract the Rn from the message and compare this received random number with the Rn that is initially forwarded to the Hn_i. In the case of Rn matches, the TS verifies the successful authentication with Hn_i otherwise, discard the received message. The *Hn* authentication and session key generation with TS is shown in algorithm 3.

Algorithm3 : Session Key generation Between Hn and TS

1. Hn_i: $Sm_{Hni} = H(\text{Rn}) \oplus H(K_{Hni})$ **then,**
2. $Hn_i \rightarrow$ TS: $Cm_{Hni} = SK_{TS_i}$ $\{(Hn_{id_i}, Ts_{Hni}, Sm_{Hni}$, $\text{H}(Cm_{Hni})\}$ **after that,**
3. TS: **If** $(Ts_{Hni}{}' - Ts_{TS} < \Delta t$ **then**
4. TS: **If** $H'(Cm_{Hni})$ equals $H(Cm_{Hni})$ **then**
5. $Sc_{Hni} \oplus Sm_{Hni}$ to extract (Rn)
6 TS: **If** $H(\text{Rn})'$ equals $H(\text{Rn})$ **then**
7 $RK_{TS-Hn} = \{\text{Rn}_1, \text{Rn}_2, \ldots \text{Rn}_r\}$
8. TS: $SK_{Hn-TS} = \text{H}(Hn_{id_i}) \oplus \text{H}(K_{Hn_i}) \oplus \text{H}(RK_{TS-Hn}) \oplus \text{H}(Rn)$
9. TS: Forwards the RK_{TS-Hn} to the Hn_i for session key generation at Hn_i
10. **Else**
11. Drop the message due data falsification
12. **End if**
13. **Else**
14. Drop the message due data falsification
15. **End if**
16. **Else**
17. Drop the message due data falsification
18. **End if**

For session key generation all head nodes $Hn_{i \ldots n}$ are prepared a message Cm_{Hni} and forwards to the TS as shown in equation 7. The message $Sm_{Hni} = H(\text{Rn}) \oplus H(K_{Hni})$ contains pre-received random numbers $H(\text{Rn}) \oplus$ with the hash of secret key $H(K_{Hni})$.

$$Cm_{Hni} = Sc_{TS} \ \{(Hn_{id_i}, Ts_{Hni}, Sm_{Hni}, \text{H}(Cm_{Hni})\} \quad (7)$$

TS receives the Cm_{Hni} message from the Hn_i and checks the difference of timestamps $(Ts_{Hni}{}' - Ts_{TS}) < \Delta$t. $Ts_{Hni}{}'$ is a sending time stamp and Ts_{TS} is a receiving timestamp. In case the variance of timestamps is under the threshold value message is verified otherwise discarded. Next, ensure message integrity by matching the similarity of receiving hash $H'(Cm_{Hni})$ from Hn_i with the calculated $H(Cm_{Hni})$ hash at TS. Furthermore, TS taking $Sc_{TS} \oplus Sm_{Hni}$ to extract Rn with the each Hn_i. Moreover, message to extract the hash of the random numbers (Rn) from the received messages of each Hn_i and compare this received Rn' with the Rn that is initially forwarded to the Hn_i. After that, TS randomly selects the $RK_{Hn-Mn} = \{\text{Rn}_1, \text{Rn}_2, \ldots \text{Rn}_r\}$ values of head nodes to form a random key RK_{TS-Hn} and forwarding this value to the Hn_i for session key generation.

$$SK_{Hn-TS} = \text{H}(Hn_{id_i}) \oplus \text{H}(K_{Hn_i}) \oplus \text{H}(RK_{TS-Hn}) \oplus \text{H}(Rn) \quad (8)$$

The session key is generated at the trusted server as given in equation 8. Moreover, Mn_i receives the hash of RK_{TS-Hn} and uses the RK_{TS-Hn} with other credentials that are forward to the TS such as $(Hn_{id_i}, K_{Hn_i}, Rn)$ to generate the same session key without sharing the key over the network.

5. Results and Analysis

This section elaborates the outcomes of SMKD and also provides an analysis of the results. Therefore, a simulation environment is implemented in the NS 2. In our simulation scenario, the area of 1500 × 1500 m is considered. AWK files are employed to attain results from trace files. Moreover, the simulation parameters are listed in Table 2.

Table 2: List of Simulation Parameters

Parameters	Values
Simulation Area	1500 × 1500 m
Group range	400 m
Initial energy of Mn	1000 J
Initial energy of Hn	10,000 J
Energy for message transmission	0.819 µJ
Energy for message receiving	0.049 µJ
Mac Protocol Type	Mac/802–11
Queue type	DropTail/PriQue
Max Packet in Queue	50
Mobile nodes in a group	5–10 nodes
Message Size	100–500 bytes
Number of Messages	5–30 messages
Given Time Slot	0.1–1.0 s
Responding Node Count	50–250 nodes

5.1 Energy Consumption

In the energy analysis, we set the initial energy of the head node as 10,000 J. The energy consumption of the head node is considered during node authentication and key generation and attain the remaining energy of the head nodes from the trace files. Moreover, we extracted the energy consumption of head nodes by taking the variance of initial and recent energy of head nodes by employing AWK files. In Figure 3(a), we represent the energy consumption while authentication of the head node. Therefore, SMKD is contrasted against counterparts. At the instance of 0.6 seconds, the energy consumption of SPPDA, PFDA, PPDAS, SMKD is 0.0063 joules, 0.0069 joules, 0.0078 joules, 0.0054 joules, respectively. In Figure 3(b), we illustrate the energy utilization of mobile nodes in the authentication process. Moreover, the mobile nodes' initial energy

is set to 1000 J. The energy consumption of sensor nodes is considered during node authentication. Furthermore, AWK files are employed to attain results by taking the variance of initial and recent values. At the instance of 0.6 seconds, the energy consumption of SPPDA, PFDA, PPDAS, SMKD is 0.0062 joules, 0.0046 joules, 0.0065 joules, 0.0057 joules, 0.0041 joules, respectively. respectively.

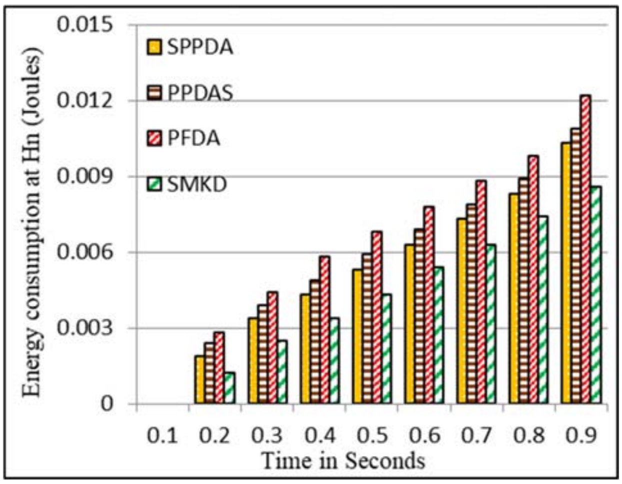

Fig 3(a) Energy Consumption at Head Node

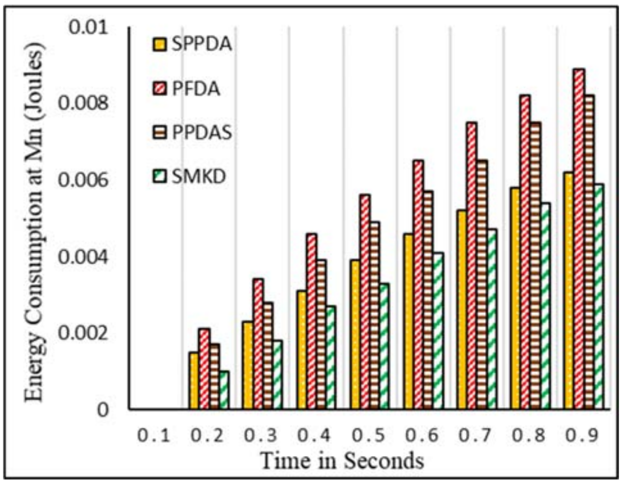

Fig 3(b) Energy Consumption at Mobile Node

5.2 Computational Cost

In the case of mobile node authentication at the head node. The number of mobile nodes sends an encrypted message that includes a pre-received random number for authentication at the head node. In Figure 4(a), we calculate the computational cost during the authentication procedure of mobile nodes. Hence, in the case of 100 mobile nodes, the cost of computation for PPDAS, PFDA, SMKD, and SPPDA is 34.93, 40.05, 12.8, and 22.8 milliseconds. The trusted server receives an encrypted message from the head node for registration and authentication. The trusted server verifies the message authentication and also compares the pre-forwarded random number for head node authentication.

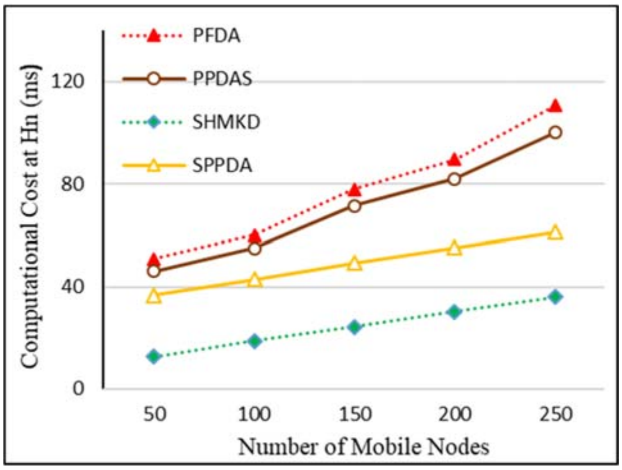

Fig 4(a) Computational Cost at Head node

In Figure 4(b) we are considering the computational cost against the number of head nodes. Hence, in the case of 100 head nodes the cost of the computational cost of the PPDAS, PFDA, SMKD, and SPPDA is 60.50 ms, 54.93 ms, 18.08 ms, and 42.61 ms. The results illustrate that SMKD provides efficient communication cost while comparing with SPPDA, PPDAS, and PFDA schemes.

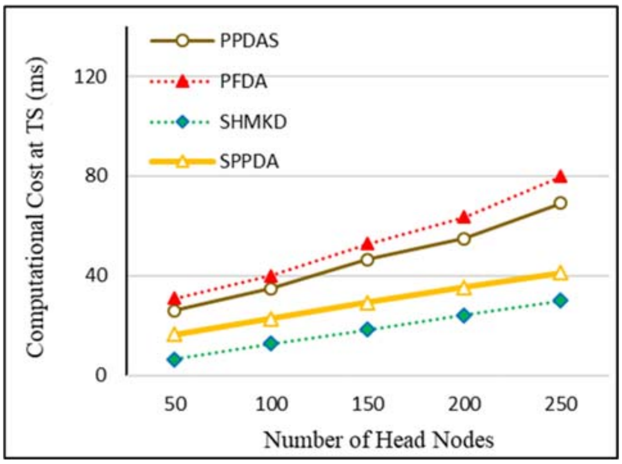

Fig 4(b) Computational Cost at Trusted Server

5.3 Communication Cost

In the case of communication cost, we are considering the communication cost during node authentication for secure communication. In this context, the communication cost is measured in two different situations such as mobile node authentication at head node, and Head node authentication at a trusted server. In Figure 5(a) we are considering the communication cost while authentication of individual mobile node at the head node. The analysis for communication cost is conducted against SPPDA, PPDAS, and PFDA schemes. In this context, the mobile node shares an authentication message with the head node. The encrypted message contains XOR of random values and timestamp of 64 bits each, 32 bits of node password, 64 bits of nuance value, 256 bits of message hash, and 64 bits of the symmetric key. Thus, the head node forwards 560 bits of an encrypted message to the head node. In the case of mobile node authentication, our proposed scheme provides better communication costs while comparing with the counterparts. Hence, in the case of 150 mobile nodes the communication cost of SMKD, SPPDA, PPDAS, PFDA is 30.54, 40.08 ms, 65.32 ms, and 73.84 ms, respectively.

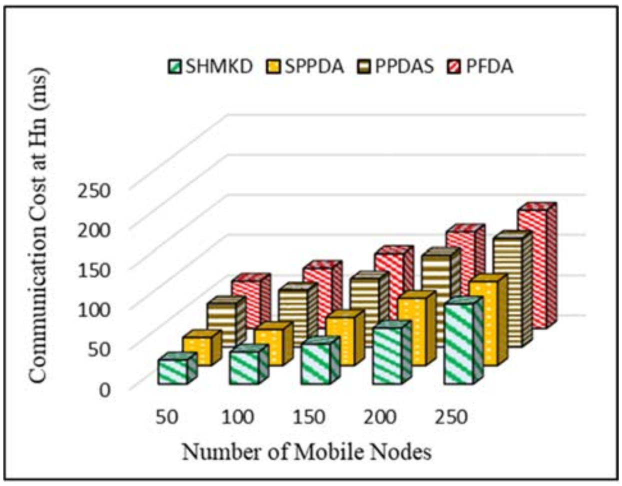

Fig 5(a) Communication Cost at Head node

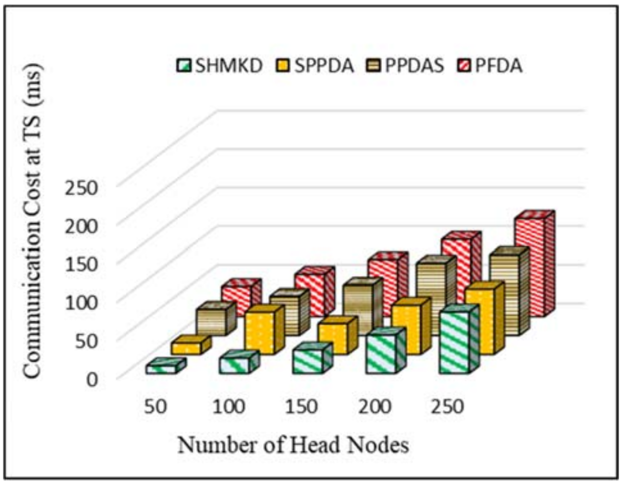

Fig 5(b) Communication Cost at Trusted Server

In Figure 5(b) we consider communication cost for authentication of the individual head node at the trusted server. In this context, the head node sends the encrypted message of 496 bits towards the trusted server for authentication. In the case of 150 head nodes, the communication cost of SMKD, SPPDA, PPDAS, PFDA is 50.02 ms, 60.93 ms, 85.29 ms, and 93.64 ms, respectively. The results prove that our proposed scheme provides better communication cost.

6. Conclusion

A Secure Multipart Key Distribution scheme (SMKD) that provides secure communication among the nodes by generating a multiparty secure key for communication. We provide node authentication and session key generation mechanism among mobile nodes, head nodes, and trusted servers. Firstly, mobile nodes are authenticated at the head node. Secondly, the head node generates a secure session key with mobile nodes and also with other head nodes. Finally, TS authenticate head nodes and also generate session keys with head nodes for secure communication. The symmetric key-based encryption method is in SMKD. Moreover, SHA 2 is employed to protect the message's integrity. We analyze SMKD in contrast with counterparts to evaluate the performance. A simulation environment is implemented in the NS 2. In the simulation, Node deployment and message initiation are employed by TCL files. Moreover, AWK scripts are employed to attain results for analysis. Results prove that SMKD provides better results against SPPDA, PFDA, and PPDAS schemes for energy consumption, communication, and computational cost. In the future, we shall evaluate the performance for real-time data transmission and key generation and also adopt some security mechanisms to further enhance the performance.

Acknowledgments

The author wishes to acknowledge the approval and the support of this research study by the grant N°CIT-2018-3-9-F-7964 from the Deanship of the Scientific Research at Northern Border University, Arar, Kingdom of Saudi Arabia.

References

[1] A. Ullah, M. Azeem, H. Ashraf, A. Alaboudi .A, M. Humayun, and N. Z. Jhanjhi, “Secure Healthcare Data Aggregation and Transmission in IoT — A Survey,” *IEEE Access*, vol. 9, pp. 16849–16865, 2021.

[2] J. Singh, R. Kaur, and D. Singh, “A survey and taxonomy on energy management schemes in wireless sensor networks,” *J. Syst. Archit.*, vol. 111, pp. 1–22, 2020.

[3] H. Mrabet, S. Belguith, A. Alhomoud, and A. Jemai, "A survey of IoT security based on a layered architecture of sensing and data analysis," *Sensors*, vol. 20, no. 13, pp. 1–19, 2020.
[4] N. Alhirabi, O. Rana, and C. Perera, "Security and privacy requirements for the Internet of Things : A survey," *ACM Trans. Internet Things*, vol. 2, no. 1, pp. 1–37, 2021.
[5] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, "Security and Privacy in the Medical Internet of Things : A Review," *Secur. Commun. Networks*, pp. 1–9, 2018.
[6] J. J. Hathaliya and S. Tanwar, "An exhaustive survey on security and privacy issues in Healthcare 4 . 0," *Comput. Commun.*, vol. 153, pp. 311–335, 2020.
[7] A. Kumar, S. Zeadally, and D. He, "Taxonomy and analysis of security protocols for Internet of Things," *Futur. Gener. Comput. Syst.*, vol. 89, pp. 110–125, 2018.
[8] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, 2017.
[9] S. Al-Janabi, I. Al-Shourbaji, M. Shojafar, and S. Shamshirband, "Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications," *Egypt. Informatics J.*, vol. 18, no. 2, pp. 113–122, 2017.
[10] H. Ahmadi, G. Arji, L. Shahmoradi, R. Safdari, M. Nilashi, and M. Alizadeh, "The application of internet of things in healthcare: a systematic literature review and classification," *Univers. Access Inf. Soc.*, vol. 18, pp. 837–869, 2018.
[11] Y. Sun, H. Zhu, and X. Feng, "A Novel and Concise Multi-receiver Protocol Based on Chaotic Maps with Privacy Protection," vol. 19, no. 3, pp. 371–382, 2017.
[12] A. Ara, M. Al-Rodhaan, Y. Tian, and A. Al-Dhelaan, "A Secure Privacy-Preserving Data Aggregation Scheme Based on Bilinear ElGamal Cryptosystem for Remote Health Monitoring Systems," *IEEE Access*, vol. 5, pp. 12601–12617, 2017.
[13] M. Farash, Sabzinejad, M. Attari, Ahmadian, and S. Kumari, "Cryptanalysis and improvement of a three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps," *Int. J. Commun. Syst.*, vol. 30, no. 1, pp. 1–10, 2014.
[14] S. Chiou and C. Lin, "An Efficient Three-Party Authentication Scheme for Data Exchange in Medical Environment," *Secur. Commun. Networks*, vol. 2018, pp. 1–15, 2018.
[15] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. P. C. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4755–4763, 2019.
[16] J. O. Kwon, I. R. Jeong, and D. H. Lee, "Practical Password-Authenticated Three-Party Key Exchange," *KSII Trans. Internet Inf. Syst.*, vol. 2, no. 6, pp. 977–980, 2008.
[17] C. Lu, "Multi-party Password-Authenticated Key Exchange Scheme with Privacy Preservation for Mobile Environment," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 12, pp. 5135–5149, 2015.
[18] C. Guo, P. Tian, and K.-K. R. Choo, "Enabling Privacy-assured Fog-based Data Aggregation in E-healthcare Systems," *IEEE Trans. Ind. Informatics*, vol. 17, no. 3, pp. 1948–1957, 2020.
[19] 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," *Comput. Networks*, vol. 129, pp. 429–443, 2017.
[20] H. Liu, X. Yao, T. Yang, and H. Ning, "Cooperative privacy preservation for wearable devices in hybrid computing-based smart health," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1352–1362, 2018.
[21] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of Secure User Authenticated Key Management Protocol for Generic IoT Networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, 2018.
[22] M. Wazid and S. Member, "A Tutorial and Future Research for Building a Blockchain-Based Secure Communication Scheme for Internet of Intelligent Things," *IEEE Access*, vol. 8, no. 1, pp. 88700–88716, 2020.
[23] X. Shen, L. Zhu, C. Xu, K. Sharif, and R. Lu, "A privacy-preserving data aggregation scheme for dynamic groups in fog computing," *Inf. Sci. (Ny).*, vol. 514, pp. 118–130, 2020.
[24] J. Gowthami, "Secure Three-Factor Remote user Authentication for E-Governance of Smart Cities," *2018 Int. Conf. Curr. Trends Towar. Converging Technol.*, no. May, pp. 1–8, 2019.
[25] X. Jia and D. He, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wirel. Networks*, vol. 25, pp. 4737–4750, 2018.
[26] H. Zhong, L. Shao, J. Cui, and Y. Xu, "An efficient and secure recoverable data aggregation scheme for heterogeneous wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 111, pp. 1–12, 2018.
[27] O. R. Merad Boudia, S. M. Senouci, and M. Feham, "Secure and efficient verification for data aggregation in wireless sensor networks," *Int. J. Netw. Manag.*, vol. 28, no. 1, pp. 1–17, 2018.

Saleh Altowaijri obtained his PhD from Swansea University in the area of cloud computing. Currently he is an Associate Professor at the Department of Information Systems, Faculty of Computing and Information Technology, Northern Border University, Rafha, Kingdom of Saudi Arabia. He has over 10 years of research experience and has published several book chapters, conference and journal papers. He is a reviewer of several international conferences and journals. His research interests include grid and cloud computing, database management systems, data mining, information systems, information technology risk management, IoT, and emerging ICT systems in healthcare and transportation sectors.