LSSCW: A Lightweight Security Scheme for Cluster based Wireless Sensor Network

Ganesh R. Pathak[1]
1. PhD Scholar,
Department of Computer Science and Engineering,
Satyabhama Institute of Science and Technology
(Deemed to be University),
Jeppiaar Nagar, Rajiv Gandhi Road,
Chennai, Tamilnadu, India 600119
2. Sinhgad College of Engineering,
Vadgaon (Bk), Pune 411041

M.S.Godwin Premi[2]
School of Electrical and Electronics Engineering,
Satyabhama Institute of Science and Technology
(Deemed to be University),
Jeppiaar Nagar, Rajiv Gandhi Road,
Chennai, Tamilnadu, India 600119

Suhas H. Patil[3]
Department of Computer Engineering,
Bharati Vidyapeeth (Deemed to be University),
Pune-Satara Road, Katraj,
Pune, Maharashtra, India 411043

***Abstract*—In last two decades, Wireless Sensor Network (WSN) is used for large number of Internet of Things (IoT) applications, such as military surveillance, forest fire detection, healthcare, precision agriculture and smart homes. Because of the wireless nature of communication, Wireless Sensor Network suffers from various attacks such as Denial of Service (DoS) attack and replay attack. Dealing with scalability and security issues is the challenging task in WSN. In this paper, we have presented a Lightweight Security Scheme for Cluster based Wireless Sensor Network (LSSCW). LSSCW has two phases: Initialization phase and data transfer phase. The work focuses on secured data aggregation in wireless sensor network with the help of symmetric and session key generation technique. Data from sensor nodes are securely transferred to base station. LSSCW is lightweight and satisfies security requirements including authenticity, confidentiality and integrity. The performance of LSSCW is verified using Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. Results shows that LSSCW is secured and is efficient in terms of computation and communication overhead.**

Keywords*—Authentication; Automated Validation of Internet Security Protocols and Applications tool; Internet of Things (IoT); key management; security; Wireless Sensor Network (WSN)

I. Introduction

Wireless Sensor Network (WSN) is a collection of large number of sensor devices which cooperatively work with each other for monitoring the environmental conditions [1], [2]. Limited energy and limited computational capability of Sensor Nodes (SN) make the WSN critical compared to traditional networks.

Covering large geographical area needs numerous sensor nodes which introduces the scalability issue while developing the routing solutions for WSN [3].

The cluster based architecture [4], as shown in Fig. 1, able to handle the scalability issue due to distributed control of the sensor nodes in the network. Each cluster is controlled by a special node having more computational capability and energy compared to sensor nodes; referred as Cluster Head (CH). CH collects the information from the sensor nodes under its cluster, if required, aggregates the data and sends the aggregated data to the Base Station (BS). BS is the final destination node where all the sensor nodes' data is collected and used for further analytic. Large number of applications are developed based on the WSN which ranges from applications such as military/battlefield surveillance, forest fire detection, health care, precision agriculture, smart homes and smart grid [5] [6].

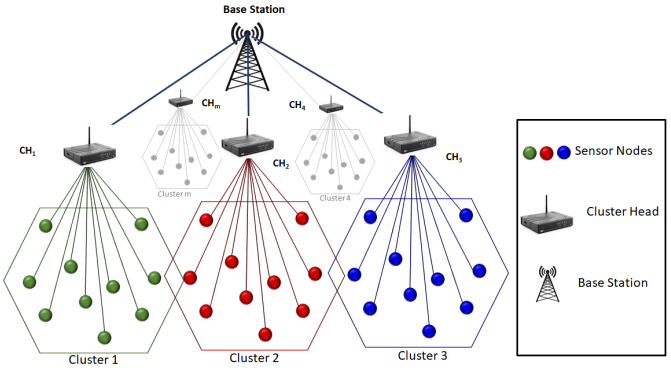

Fig. 1. Cluster based Wireless Sensor Network

Similar to computer networks, WSN faces the security concerns [7] challenging integrity, confidentiality and authenticity of nodes and data. Data stored at legitimate nodes and the data communicated over the wireless channel must be protected from the attacker. The passive attack by the attacker breaks the confidentiality of the communicating parties and the data whereas the active attack by the attacker raises the

question regarding integrity and authenticity of the messages. The intensity of these attacks varies based on the applications where the WSN is used. Thus, the security becomes critical requirement in WSN.

In this paper, we presented the security solution for cluster based WSN which satisfies security requirements including authentication of communicating parties, confidentiality and integrity of data. The network model consists of track and sector architecture [8] [9] to introduce various clusters in the network and each cluster covers number of sensor nodes. To control the sensor nodes, each cluster has two special nodes namely; Data Cluster Head (DCH) and Routing Cluster Head (RCH). DCH is responsible for collecting data from all sensor nodes under respective cluster, aggregating the data and sending the aggregated data to RCH. RCH is responsible for routing the data received from DCH to BS. Thus, the data by all sensor nodes is collected at BS. During the communication, data is securely communicated to other party using our proposed scheme. Our major contributions are as follows:

1) Communicating parties are mutually authenticated and the lightweight solution is provided for key generation between communicating parties.
2) Lightweight security scheme is proposed which provides end-to-end security and satisfies security requirements including authenticity, confidentiality and integrity.
3) The proposed scheme is simulated using Automated Validation of Internet Security Protocols and Applications(AVISPA) tool and the results show that the scheme is secured.

The paper is organized as follows. Section II presents related work. Section III presents the system model for the proposed work which covers network model, adversary model and security requirements and security goals. Section IV gives details of the proposed Lightweight Security Scheme for Cluster based Wireless Sensor Network (LSSCW). Security analysis is discussed in Section V. Security verification using AVISPA is presented in Section VI. Performance evaluation is done Section VIIfollowed by the conclusion in Section VIII.

II. Related Work

Key generation and distribution plays an important role in secured data transfer in WSN. Initially, the methods are developed for secure data transfer between communicating parties in WSN. In recent years, WSN is used for Internet of Things (IoT) applications which has introduced new methods for key generation concentrating on specific applications.

Saraswathi et al. [10] presented the multi-stage key management scheme for cluster based WSN. It has three stages including pre-deployment of required parameters, key generation and key authentication and verification. After cluster formation, each node is loaded with the predefined network key. In the key generation stage, keys are generated for sensor nodes, CH and BS using GM Encryption scheme [11]. The third stage consists of key authentication and verification between sender and receiver before sending every data message. This work has major two limitations. As all nodes are preloaded with the same network key, it becomes harmful if an attacker recognizes this key from a single node. The second limitation is communication overhead incurred for authentication of sender and receiver in the third stage. Jiang et al. [12] proposed the Rabin cryptosystem based authentication and key agreement protocol. Even though it has advantage from security perspective, it needs more computation overhead at gateway node compared to Das's protocol and Amin et al.'s protocol. Athmani et al. [13] solved the key distribution problem but not secured against replay attack and insider attack. Turkanovic et al. [14] invented the authentication scheme which is lightweight in terms of computation overhead but faces security weaknesses as per [15]–[17].

Key management in dynamic environment was also the focus of number of researchers. Vaneeta and Kumar [18] presented a key generation scheme for dynamic network of WSN. The work presents multiple layers of security using lightweight cryptography to secure the system against key-based attacks. Various key pre-distribution schemes are presented in [19]–[22]. Hu and Gharavi [19] used merkle tree for multi way handshaking during dynamic key distribution process. Choi et al. [20] used eighenvectors for securing the WSN from malicious tampering of the secret keys. Bag and Roy [21] presented the key pre-distribution scheme for general and grid-group deployment in WSN whereas Bechkit et al. [22]proposed the key distribution scheme for scalable WSN. In 2016, Pathak and Patil [23], we have presented the key distribution protocol for mobile WSN. The protocol was based on one-way hash function and exclusive-or operation.

Elliptic Curve Cryptography (ECC) was the choice of many researchers to develop the mutual authentication and key establishment protocol [15], [24]–[27]). As the ECC provides same level of security compared to traditional techniques such as RSA with reduced key size and simple computations, these protocols show better performance in terms of computation overhead. Khan et al. [28] presented ECC based mutual authentication and key establishment protocol where different classes of nodes can authenticate each other and establish the secure communication. Qin et al. [29] presented the key management scheme for scalable network where Elliptic Curve Paillier encryption [30] is used for communication and AVL tree is used to store the neighbors' ID and public key which reduces the search time. Node addition and deletion in the network was supported by the scheme. Nadir et al. [31] used ECC for generation of pairwise symmetric keys. Recently, Agarkar and Agrawal [32] presented the Password Authenticated Key Exchange by Juggling (J-PAKE) and ECC based authentication protocol for smart grid which is designed for mutual authentication and key generation between communicating parties in smart grid. Lattice cryptography based security and privacy preservation schemes are developed for smart grid network which includes key generation and secured data transfer for smart grid [33], [34]. Shen et al. [35] has introduced key generation and authentication protocol for wireless body area networks (WBANs). The protocol was developed based on ECC and message authentication code (MAC). The limitation the work is the protocol was vulnerable to replay and impersonation attack.

Wireless sensor network is used in Internet of Things applications. El-hajj et al. [36] has presented the review on various authentication schemes which are developed for IoT

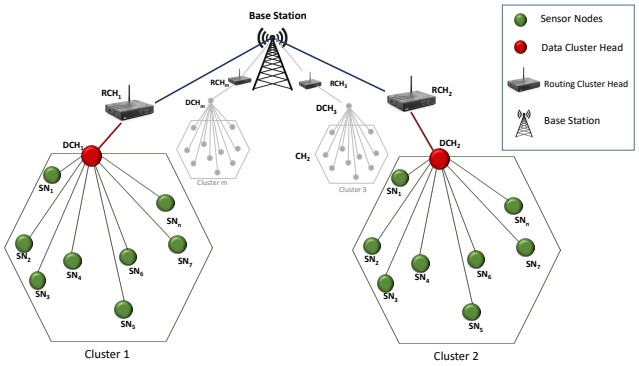

Fig. 2. Proposed Cluster based Wireless Sensor Network

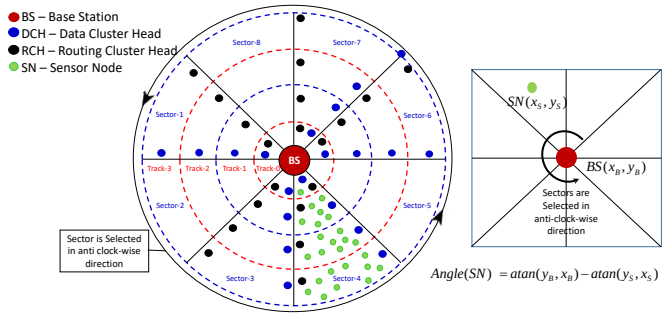

Fig. 3. Logical Track and Sector Mechanism for Cluster Formation

applications using WSN. Recently, Al-Zubaidie et al. [37] presented the user authentication scheme for health care application to protect patient's data from internal and external attackers. It uses Elliptic Curve Integrated Encryption Scheme (ECIES) and PHOTON for protecting data against malicious users. Performance of the proposed work was checked on AVISPA tool.

The above literature review indicates that the security mechanism increases both, the communication and computation overhead. The schemes which provides high security are heavy in terms of computation overheads. On the other hand, other schemes are lightweight but weak in terms of security concern and suffers from various attacks. Therefore, a lightweight solution minimizing theses overheads is an open research problem.

In this paper, we have presented the scheme consisting of key generation and secured data transfer mechanism which is lightweight in terms of computation and communication overhead.

III. System Model

System model describes about the network model, adversary model and security requirements. Finally, it lists the security goals of the proposed scheme.

A. Network Model

The network consists of large number of sensor devices arranged in a cluster based topology as shown in Fig. 2. A track and sector based mechanism ([8] [9]) is used to define clusters in the network. Fig. 3 shows an example of arrangement of sensor nodes in the clusters. Dotted circles are tracks and each track is divided into 8 sectors with each sector originating at base station. A cluster is a part of the track bounded by a sector. As the objective of the work is to develop secured data aggregation, it is assumed that the node deployment strategy shall ensure that there are at least three nodes in a cluster to facilitate sensing within the cluster, single hop/multi-hop communication between the cluster heads and base station. Each cluster has sensor nodes to sense the environmental conditions, a Data Cluster Head for data aggregation and a Routing Cluster Head to perform routing decisions.All sensor nodes are informed about DCH and RCH of their respective clusters. DCH collects the sensor data from all sensors from respective cluster and aggregates the data. The aggregated data is sent to RCH which is responsible to send the data to BS. We have divided the task of aggregation and routing to balance the energy consumption for these two nodes. Our network consists of one BS, m DCHs namely, $\{DCH_1, DCH_2, ..., DCH_m\}$, m RCHs namely, $\{RCH_1, RCH_2, ..., RCH_m\}$ and n sensor nodes under each cluster. Communication between all parties is through wireless channel. It is assumed that each sensor has its unique ID and stored in secured way.

B. Adversary Model and Security Requirements

Because of the wireless nature of communication, WSN has risks from various attacks from adversary. The probable attacks in WSN are passive and active attacks. As the part of passive attack, adversary is interested to sense the data sent on the communication network whereas in the active attack, an adversary may tamper the data and the false data is sent to the receiver. The example attacks in WSN are node compromise attack, Denial of service (DoS) attack, black hole attack, sinkhole attack, selective message forwarding attack, Man-in-the-middle (MITM) attack and replay attack. To protect the WSN from these attacks, the system model must satisfy security requirements such as confidentiality, message integrity, authenticity and availability.

C. Security Goals

Our proposed scheme achieves two major goals:

- LSSCW guarantees security of all parties during the communication. Sensors' data is securely communicated till BS and achieves integrity and confidentiality of data. The scheme also takes care about the availability of communicating parties during communication.
- LSSCW is efficient in terms of computation and communication overhead.

IV. A Lightweight Security Scheme for Cluster based Wireless Sensor Network (LSSCW)

The proposed LSSCW scheme consists of two phases: Initialization phase and data transfer phase. As the part of initialization phase, symmetric and session keys are generated between communicating parties. Once the keys are generated, the data is transferred from sensor node to base station as the part of data transfer phase. The shared symmetric keys are also

TABLE I. LIST OF SYMBOLS

Symbol	Description
BS	Base station
C_{DCH}	Cipher text computed by DCH
C_{RCH}	Cipher text computed by RCH
C_S	Cipher text computed by sensor node
DCH	Data cluster head
$h(.)$	Hash function
$\oplus$	Exclusive OR operation
$\|\|$	Concatenation operation
ID_S, ID_D, ID_R, ID_B	Identity of sensor node, DCH, RCH and BS respectively
k_{DR}	Shared key between DCH and RCH
k_{RB}	Shared key between RCH and BS
k_{SD}	Shared key between sensor node and DCH
M	Data generated at sensor node
n_B	Nonce generated by BS
n_D	Nonce generated by DCH
n_R	Nonce generated by RCH
PSW_d	Shared password between DCH nd RCH
PSW_r	Shared password between BS and RCH
PSW_s	Shared password between Sensor node and DCH
RCH	Routing cluster head
S	Sensor node
SK	Session key between DCH and BS
T_1 to T_x	Time stamp
TS_{dr}	Transaction sequence number between DCH and RCH
TS_{rb}	Transaction sequence number between RCH and BS
TS_{sd}	Transaction sequence number between sensor node and DCH
V_1 to V_4	Verification values

refreshed in this phase. Table I shows the list of symbols used in this scheme.

1) *Phase I: Initialization Phase:* Initialization phase is initiated by BS. After defining the clusters and respective DCH and RCH for each cluster, BS initiates the process of shared secret key generation. Fig. 4 shows the symmetric key generation process whereas Fig. 5 shows the session key generation process. During communication, BT_1, BT_2 are the temporary values computed by BS, RT_1, RT_2 are the temporary values computed by RCH and DT_1, DT_2, AID_D are the temporary values computed DCH.
   a) Shared symmetric key generation:
      - *Message 1(BS $\rightarrow$ RCH):* BS has the shared password with RCH which is PSW_r. BS generates the nonce n_B and computes $BT_1 = h(ID_R||PSW_r||ID_B) \oplus n_B$. BS also maintains the transaction sequence number for RCH. Consider TS_{rb} is the current transaction sequence number. Using this TS_{rb}, BS computes, $BT_2 = h(ID_R||n_B) \oplus TS_{rb}$. BS sends the message containing (BT_1, BT_2) to RCH. After receiving the message, RCH tries to find the values of the nonce generated by BS and the transaction sequence number. RCH computes $n_B = h(ID_R||PSW_r||ID_B) \oplus BT_1$ and $TS_{rb} = h(ID_R||n_B) \oplus BT_2$. Now RCH and BS computes the shared key $k_{RB} = h(BT_1||n_B||TS_{rb})$.
      - *Message 2(RCH $\rightarrow$ DCH):* RCH generates PSW_d and computes $PSW_d^* = h(ID_D||ID_R) \oplus PSW_d$. RCH maintains the record of transaction sequence number TS_{dr} for communication with DCH. RCH generates the nonce n_R and computes $RT_1 = h(ID_D||PSW_d||ID_R) \oplus n_R$ and $RT_2 = h(ID_D||n_R) \oplus TS_{dr}$. RCH sends (PSW_d^*, RT_1, RT_2) to DCH. DCH finds password as, $PSW_d = h(ID_D||ID_R) \oplus PSW_d^*$, the nonce n_R as $n_R = h(ID_D||PSW_d||ID_R) \oplus RT_1$ and transaction sequence number TS_{dr} as $TS_{dr} = h(ID_D||n_R) \oplus RT_2$. RCH and DCH computes the shared key as, $k_{DR} = h(RT_1||n_R||TS_{dr})$.
      - *Message 3 (DCH $\rightarrow$ sensor node):* DCH starts the process of key generation with sensor node. DCH generates a password PSW_s and computes, $PSW_s^* = h(ID_S||ID_D) \oplus PSW_s$. DCH generates nonce n_D and computes $DT_1 = h(ID_S||PSW_s||ID_D) \oplus n_D$. DCH maintains the sequence num-

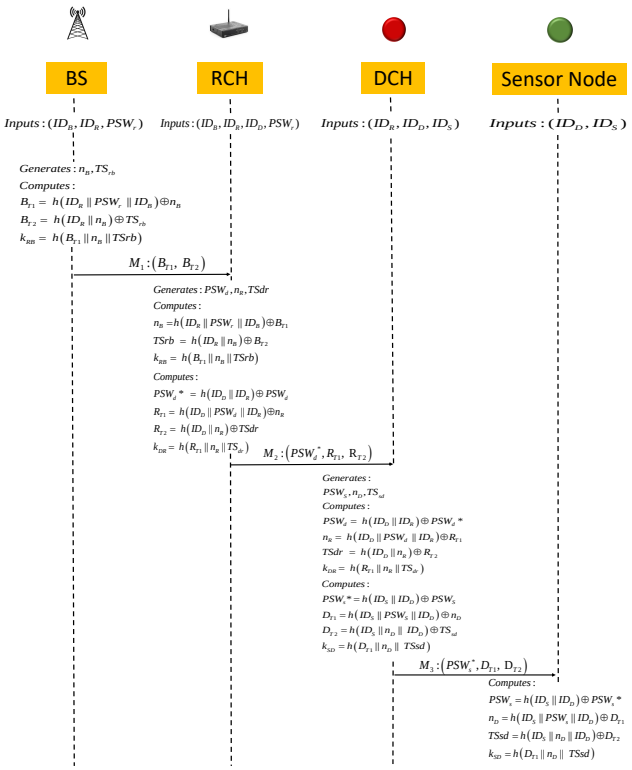

Fig. 4. Phase I: Symmetric Key Generation

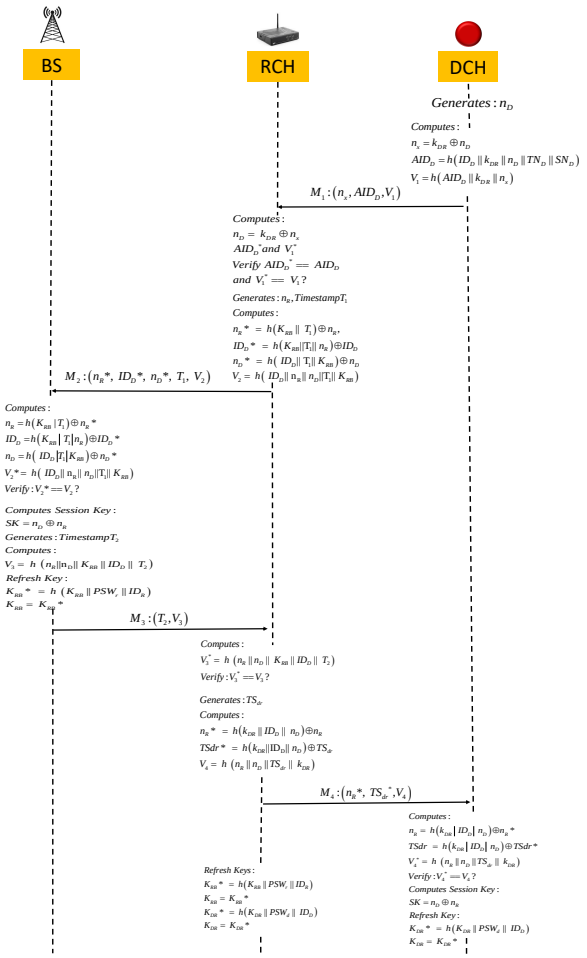

Fig. 5. Phase I: Session Key Generation

ber for each sensor node. TS_{sd} is the recent transaction sequence number for specific sensor node. DCH computes $DT_2 = h(ID_S||n_D||ID_D) \oplus TS_{sd}$. DCH sends the message containing (PSW_s^*, DT_1, DT_2) to sensor node. DCH maintains the record of shared password, nonce and transaction sequence number for each sensor node in its database.

After receiving the message, sensor node computes, $PSW_s = h(ID_S||ID_D) \oplus PSW_s^*, n_D = h(ID_S||PSW_s||ID_D) \oplus DT_1$ and $TS_{sd} = h(ID_S||n_D||ID_D) \oplus DT_2$. Now both parties computes the shared secret key as $k_{SD} = h(DT_1||n_D||TS_{sd})$. Sensor node uses k_{SD} for securely sending data to DCH during data transfer phase.

b) *Shared session key generation:* Once the pairwise keys are generated, DCH starts the process of session key generation between DCH and BS. This key helps in reducing the computation overhead during data transfer phase. The session key is generated with the help of nonce generated at DCH and RCH. Following is the process for generating the key.

- *Message 1 (DCH $\rightarrow$ RCH):* DCH generates a nonce n_D. This nonce need to be send to BS with the help of RCH in secured way. Hence, DCH computes $n_x = k_{DR} \oplus n_D$. DCH also generates the one time alias identity as $AID_D = h(ID_D||k_{DR}||n_D||TN_D||SN_D)$ and verification value $V_1 = h(AID_D||k_{DR}||n_x)$. DCH sends (AID_D, n_x, V_1) to RCH. After receiving the message, RCH finds the value of nonce as $n_D = k_{DR} \oplus n_x$. RCH computes AID_D and V_1 value at its end and compares with the received entries. If verification holds, RCH considers the n_D as the valid value.
- *Message 2 (RCH $\rightarrow$ BS):* RCH generates a nonce n_R and sends n_D and n_R values to BS in secured way. RCH computes

$$n_R^* = h(k_{RB}||T_1) \oplus n_R \quad (1)$$
$$ID_D^* = h(k_{RB}||T_1||n_R) \oplus ID_D \quad (2)$$
$$n_D^* = h(ID_D||T_1||k_{RB}) \oplus n_D \quad (3)$$
$$V_2 = h(ID_D||n_R||n_D||T_1||k_{RB}) \quad (4)$$

Where T_1 is time stamp and V_2 is the verification value which will be used at BS. RCH sends $(n_R^*, ID_D^*, n_D^*, T_1, V_2)$ to BS.

BS receives the message from RCH and finds the nonce values generated by DCH and RCH. The BS computes,

$$n_R = h(k_{RB}||T_1) \oplus n_R^* \quad (5)$$
$$ID_D = h(k_{RB}||T_1||n_R) \oplus ID_D^* \quad (6)$$
$$n_D = h(ID_D||T_1||k_{RB}) \oplus n_D^* \quad (7)$$

and verify the computed value by calculating $V_2^* = h(ID_D||n_R||n_D||T_1||k_{RB})$. If V_2^* and V_2 are same, BS considers the computed values as valid and computes the session key with DCH as,
$SK = n_D \oplus n_R$

- BS should acknowledge RCH that it has received the correct values of n_D and n_R. BS computes the verification value $V_3 = h(n_R||n_D|k_{RB}||ID_D||T_2)$ where T_2 is the time stamp and it sends (V_3, T_2) to RCH.

BS also refresh the shared key with RCH as, $k_{RB}^* = h(k_{RB}||PSW_r||ID_R)$
$k_{RB} = k_{RB}^*$

RCH receives the message from BS, computes V_3 and confirms that BS has correctly received nonce values.

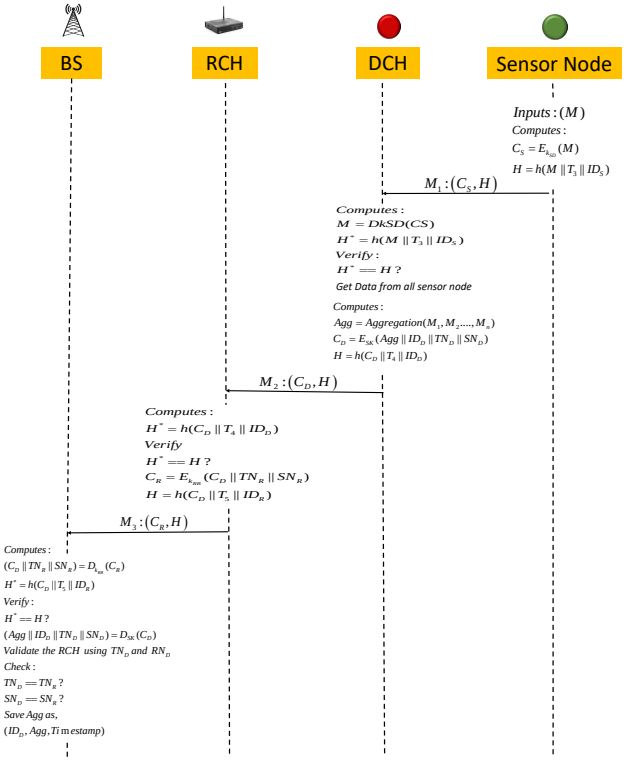

Fig. 6. Phase II: Data transfer phase

- Finally, RCH sends nonce n_R to DCH. RCH computes $n_R^* = h(k_{DR}||ID_D||n_D) \oplus n_R, TS_{dr}^* = h(k_{DR}||ID_D||n_D) \oplus TS_{dr}$ and $V_4 = h(n_R||n_D||TS_{dr}||k_{DR})$ where TS_{dr} is the current transaction sequence number. RCH sends (n_R^*, V_4, TS_{dr}^*) to DCH.
  RCH also refresh the shared keys with BS and DCH. RCH computes,
  $k_{RB}^* = h(k_{RB}||PSW_r||ID_R)$
  $k_{RB} = k_{RB}^*$
  $k_{DR}^* = h(k_{DR}||PSW_d||ID_D)$
  $k_{DR} = k_{DR}^*$
  After receiving the message, DCH computes $n_R = h(k_{DR}||ID_D||n_D) \oplus n_R^*$ and finds the new transaction sequence number as $TS_{dr} = h(k_{DR}||ID_D||n_D) \oplus TS_{dr}^*$. DCH verify these values using V_4. DCH computes the session key for communication with BS as $SK = n_D \oplus n_R$. DCH also refreshes the shared key k_{DR} with DCH as,
  $k_{DR}^* = h(k_{DR}||PSW_d||ID_D)$
  $k_{DR} = k_{DR}^*$

2) *Phase II: Data transfer phase:* Fig. 6 shows the messaging in data transfer phase. In data transfer phase sensor node senses the information and send the generated data in encrypted form to DCH. DCH decrypts the data with the shared symmetric key with respective sensor nodes, aggregates the data of all sensor nodes under its cluster. The aggregated data is encrypted using the session key between DCH and BS and forwarded to RCH. RCH again encrypts the information using its shared key with BS and finally data reaches to BS. BS performs decryption operations and recognizes the aggregated data sent by DCH. Following are the steps while sending the messages from sensor node till BS.

- *Message 1 (Sensor node $\rightarrow$ DCH):* Sensor node generates the data M. Sensor node encrypts the data using symmetric key of DCH as $C_S = E_{k_{SD}}(M)$. It also computes the hash over the data M, node's ID ID_s and time stamp T_3 as $H = h(M||T_3||ID_s)$. The encrypted data along with the hash value (C_S, H) is sent to the DCH.
  After receiving the message from sensor node, DCH decrypts the message as $M = D_{k_{SD}}(C_S)$ and check the validity using hash value H. DCH computes the hash value over the data M, node's ID ID_s and time stamp T_3. If the computed hash value and received hash value from sensor node matches, DCH saves the data M in its database.
- *Message 2 (DCH $\rightarrow$ RCH):* After collecting and decrypting data of all sensor nodes, DCH performs the aggregation operation over the received data. The aggregation function can be summation, average, finding maximum or minimum value. Choosing the aggregation function depends on the application in which the WSN is used. Consider Aggregation is the Aggregation function. DCH computes,

$$Agg = Aggregation(M_1, M_2, ..., M_n) \tag{8}$$

  Where,
  Agg is the aggregated value
  $M_1, M_2, ..., M_n$ are the data values received from sensor nodes

  The aggregated value is encrypted using the shared session key of DCH with BS. DCH performs $C_D = E_{SK}(Agg||ID_D||TN_D||SN_D)$. It also computes hash value $H = h(C_D||T_4||ID_D)$ where T_4 is the time stamp. DCH sends (C_D, H) to RCH.
- *Message 3 (RCH $\rightarrow$ BS):* RCH receives (C_D, H). RCH verifies the value of C_D by computing hash as $H^* = h(C_D||T_4||ID_D)$. If H^* and received H values matches, then RCH considers C_D as the valid value. RCH further encrypts the cipher text, C_D , using shared key between RCH and BS and finds its cipher text, $C_R = E_{k_{RB}}(C_D||TN_R||SN_R)$ and finds the hash $H = h(C_D||T_5||ID_R)$. The computed values (C_R, H) are sent to BS.
  BS receives the message and decrypts it by k_{RB} as, $(C_D, TN_R, SN_R) = D_{k_{RB}}(C_R)$.

It verifies the value of CD using received hash value. BS further decrypts C_D using session key between DCH and BS as $(Agg||ID_D||TN_D||SN_D) = E_{SK}(C_D)$. BS checks the ID of DCH from its database and confirms that the particular DCH is related to track number TN_D and SN_D. BS also checks the received track number and sector number of DCH and RCH are same. If track number of DCH, TN_D, is same as track number of RCH, TN_R, and the sector number of DCH, SN_D, is same as the sector number of RCH, SN_R then BS confirms that it has received valid aggregated value from legitimate DCH with ID_D. BS saves Agg value received from DCH in its database as $(ID_D, Agg, Timestamp)$ and can further use it for analytic purpose.

V. Security Analysis

Security analysis demonstrates that the proposed scheme preserves the privacy to individual sensor data and holds security properties which are necessary during the communication in WSN.

1) *Privacy Preservation:* In the proposed scheme, individual sensor's data is sent only to the respective DCH. DCH aggregates data from all sensor nodes and then send the aggregated data towards BS. Thus individual sensor's data is not available to RCH and BS. It helps in preserving the privacy of the sensor's data and reduces the possibility of hacking individual sensor's record.
2) *Confidentiality:* In data transfer phase, all the messages are encrypted using respective keys. Sensor node encrypts the data using shared key with DCH, k_{SD}. DCH uses the session key, SK, for encrypting the data whereas RCH uses the shared secret key between RCH and BS. Thus all messages are securely communicated to receiver nodes.
3) *Mutual Authentication:* In the proposed scheme, no third party is involved during key generation process. All the keys are generated mutually within the communicating parties. During pairwise key generation process, verification of the message is done using the hash values based on ID, password and Transaction sequence number. As the part of session key generation between DCH and BS, the one-time identity value, AID_D and the verification values V_1 to V_4 helps in verification of the messages at receiver end. During data transfer phase, the data is encrypted using respective keys and receiver authenticates the received data based on the shared key, ID and time stamp.
4) *Fair Key Agreement:* In fair key agreement scheme, each participant contribute to generate the session key. It ensures that not an individual party has unfair advantage to control the session key. In the proposed scheme, the session key between DCH and BS is generated with the help of nonce generated by DCH and RCH. DCH generates the nonce nD and RCH generates the nonce n_R. The session key is computed by DCH and BS as $SK = n_D \oplus n_R$. Thus DCH and RCH has fair involvement in the process of session key generation.
5) *Data Integrity:* In the key generation process, the nonce and transaction sequence numbers are sent to the receiver node. Integrity of the received nonce and transaction sequence numbers are verified based on ID and passwords between sender and receiver node. As the part of data transfer phase, DCH computes the cipher text CD by encrypting the sensed data M using session key between DCH and BS. It also computes hash over the cipher text CD, Time stamp and ID of DCH. RCH use this hash value to check the integrity of the C_D at its end. When RCH forwards the C_D to BS, the verification of the message is done based on hash over ID of RCH and Time stamp.
6) *Data Freshness:* Data freshness ensures that the data received by the receiver is the fresh and not the old one. It helps in resisting the replay attack. As the part of replay attack the attacker can resend the messages number of times. The proposed scheme takes care about verification of the data freshness using transaction sequence number and time stamp during message transfers. As the part of key generation phase, the transaction sequence number helps in ensuring the data freshness. During data transfer phase, the time stamp is involved in message transfer which helps in checking the freshness of data.

VI. Security Verification using AVISPA

The proposed scheme contains initialization phase and data transfer phase. Initialization phase consists of messages which are required for generation of symmetric keys between different parties and session key is also generated generated between DCH and RCH. Data transfer phase is designed for sending actual data from sensor node till base station in secured way. We have simulated the initialization phase using Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. AVISPA is designed for simulation of authentication protocols and is the standard tool referred by number of researchers. It gives the information regarding whether the designed protocol is safe and gives the time requirement for execution of the protocol. AVISPA has back end analyzers for verification of the protocol. We have verified the security of our proposed scheme using On-the-fly Model-Checker (OFMC) and Constraint-Logic-based Attack Searcher (CL-AtSe) analysers under AVISPA. OFMC employs several symbolic techniques to explore the state space in a demand-driven way where as CL-AtSe applies constraint solving with simplification heuristics and redundancy elimination techniques. AVISPA provides High Level Protocol Specification Language (HLPSL) for designing the code. We have developed the code for symmetric key generation and session key generation processes of the initialization phase.

A. Symmetric Key Generation using AVISPA

Base station, RCH, DCH and sensor node are involved in generation of symmetric keys. Pairwise keys are generated between consecutive parties. Fig. 7, Fig. 8, Fig. 9 and Fig. 10

```
%Symmetric Key geneartion: Role of Base Station (BS)
role basestation(R,B            :agent,
          PSWr                  :text,
          Hsh                   :hash_func,
  Snd,Rcv                       :channel(dy))
played_by B
def=
local State                     :nat,
TSrb,NB,BT1,BT2                 :text,
KRB                             :symmetric_key
%constants
const sec_K_RB                  :protocol_id
init State :=0
transition
1. State=0          /\ Rcv(start) =|>
  State' :=1        /\ NB' := new()
                    /\ BT1' := xor(Hsh(R.PSWr.B),NB')
                    /\ BT2' := xor(Hsh(R.NB'),TSrb)
                    /\ KRB' := Hsh(BT1'.NB'.TSrb)
                    /\ Snd(B.BT1'.BT2'.R)
                    /\ witness(B,R,k_br, KRB')
                    /\ secret(KRB', sec_K_RB,{B,R})
end role
```

Fig. 7. Symmetric key generation: Role specification of Base Station in HLPSL

```
%Symmetric Key generation: Role of Routing Cluster Head (RCH)
role rch(  D,R,B                :agent,
           PSWr                 :text,
           Hsh                  : hash_func,
           Snd,Rcv              :channel(dy))
played_by R
def=
local State                     :nat,
PSWd,TSdr,NR                    :text,
NB,TSrb,RT1,RT2,BT1,BT2         :text,
PSWdT                           : text,
KRB, KDR                        :symmetric_key
const sec_K_DR                  :protocol_id
init State :=0
transition
1. State=0          /\ Rcv(B.BT1'.BT2'.R)=|>
  State':=1         /\ NB':= xor(Hsh(R.PSWr.B),BT1') /\ TSrb' := xor(Hsh(R.NB'),BT2')
                    /\ KRB' := Hsh(BT1.NB'.TSrb')  /\ PSWd' := new()
                    /\ PSWdT' := xor(Hsh(D.R),PSWd) /\ NR' := new()
                    /\ RT1' := xor(Hsh(D.PSWd.R),NR')  /\ TSdr' := new()
                    /\ RT2' := xor(Hsh(D.NR'), TSdr') /\ KDR' := Hsh(RT1'.NR'.TSdr')
                    /\ Snd(R.PSWdT'.RT1'.RT2'.D) /\ witness(R,D,k_rd, KDR')
                    /\ secret(KDR', sec_K_DR,{R,D})
end role
```

Fig. 8. Symmetric key generation: Role specification of Routing Cluster Head in HLPSL

```
%Symmetric Key generation: Role of Data Cluster Head (DCH)

role dch(  S,D,R                :agent,
           Hsh                  : hash_func,
           Snd,Rcv  :channel(dy))
played_by D
def=
local State                     :nat,
PSWs,ND,TSsd                    :text,
PSWd,PSWdT,NR,TSdr              :text,
PSWsT, DT1, DT2, RT1, RT2       :text,
KDR, KSD:symmetric_key
const sec_K_SD                  :protocol_id
init State :=0
transition
1. State=0          /\ Rcv(R.PSWdT'.RT1'.RT2'.D) =|>
  State':=1         /\ PSWd' := xor(Hsh(D.R),PSWdT') /\ NR' := xor(Hsh(D.PSWd'.R),RT1')
                    /\ TSdr' := xor(Hsh(D.NR'), RT2') /\ KDR' := Hsh(RT1'.NR'.TSdr')
                    /\PSWs' := new()    /\ PSWsT' := xor(Hsh(S.D),PSWs')
                    /\ ND' := new()     /\ DT1' := xor(Hsh(S.PSWs.D),ND')
                    /\ TSsd' := new()   /\ DT2' := xor(Hsh(S.ND'.D),TSsd')
                    /\ KSD' := Hsh(DT1'.ND'.TSsd') /\ Snd(D.PSWsT'.DT1'.DT2'.S)
                    /\ witness(D,S, k_ds, KSD')    /\ secret(KSD', sec_K_SD,{D,S})
end role
```

Fig. 9. Symmetric key generation: Role specification of Data Cluster Head in HLPSL

```
%Symmetric Key geneartion: Role of Sensor node
role sensornode(S,D  :agent,
          Hsh        : hash_func,
          Snd,Rcv    :channel(dy))
played_by S
def=
local State             :nat,
PSWsT, PSWs, ND, TSsd, DT1, DT2:text,
KSD                     :symmetric_key
const sec_K_DS  :protocol_id
init State :=0
transition
1. State=0          /\ Rcv(D.PSWsT'.DT1'.DT2'.S)=|>
 State':=1          /\ PSWs' := xor(Hsh(S.D),PSWsT')
                    /\ ND' := xor(Hsh(S.PSWs'.D),DT1')
                    /\ TSsd' := xor(Hsh(S.ND'.D),DT2')
                    /\ KSD' := Hsh(DT1'.ND'.TSsd')
end role
```

Fig. 10. Symmetric key generation: Role specification of Sensor Node in HLPSL

```
role session(S,D,R,B            :agent,
             PSWr               :text,
             Hsh                : hash_func)
def=
%send, receive channels for all parties
local SB,RB, SR,RR,SD,RD,SS,RS: channel(dy)
composition
basestation(R,B, PSWr,Hsh,SB,RB) /\
rch(D,R,B,PSWr, Hsh, SR, RR) /\
dch(S,D,R,Hsh, SD, RD) /\
sensornode(S,D,Hsh, SS, RS)
end role
%-------------------------------------------------------
role environment() def=
          const k_br,k_rb, k_rd, k_dr,k_ds, k_sd :
protocol_id,
          s,d,r,b               :agent,
          krb,kdr,ksd           :symmetric_key,
          pswr                  :text,
          hsh                   :hash_func
intruder_knowledge = {s,d,r,b}
composition
session(s,d,r,b,pswr,hsh)
end role
```

```
goal
secrecy_of sec_K_RB,sec_K_DR, sec_K_SD
%secrecy_of KRB, KDR, KSD
%authentication
authentication_on k_rb
authentication_on k_dr
authentication_on k_sd
authentication_on k_br
authentication_on k_rd
authentication_on k_ds
end goal
%-------------------------------------------------------
environment()
```

Fig. 11. Symmetric key generation: Specification of session, environment and goal in HLPSL

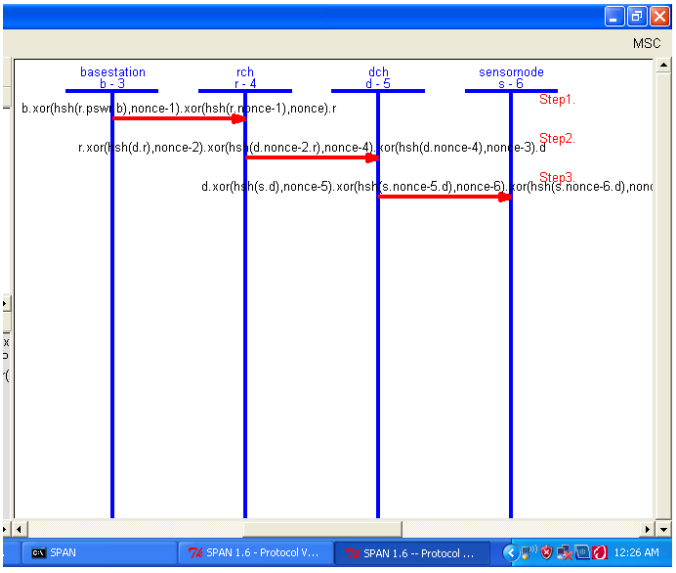

Fig. 12. Symmetric key generation: Simulation

shows the roles specified by BS, RCH, DCH and sensor node,

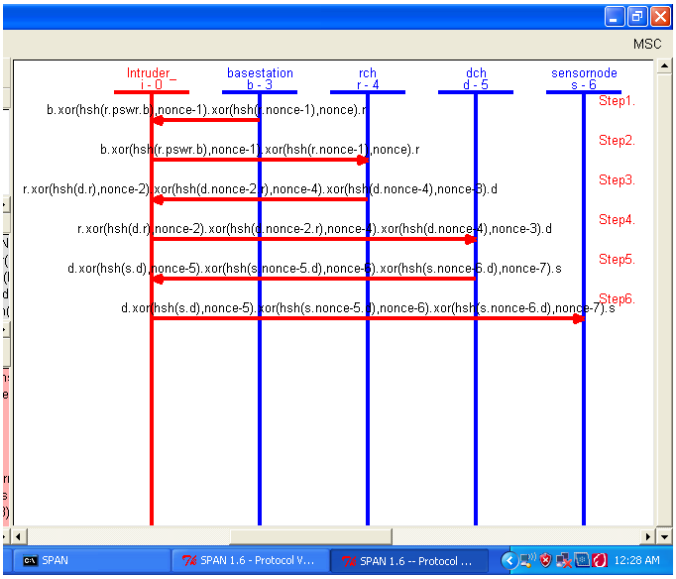

Fig. 13. Symmetric key generation: Intruder simulation

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\Symmetric.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.14s
  visitedNodes: 5 nodes
  depth: 4 plies
```

(a)

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  C:\progra~1\SPAN\testsuite\results\Symmetric.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 6 states
  Reachable  : 3 states
  Translation: 0.03 seconds
  Computation: 0.01 seconds
```

(b)

Fig. 14. Simulation results of Symmetric key generation: (a) OFMC model, (b) CL-AtSe model

respectively. Each node works in state cycle and when an event occurs, the specified action is done by respective party. Fig. 11 defines the specification of session, environment and goal in HLPSL code.

The symmetric key generation process is simulated using AVISPA. Fig. 12 shows the simulation of the messages sent between the communicating parties. Fig. 13 shows the intruder simulation. Fig. 14 shows the output of OFMC and CL-AtSe model which gives the result as the designed work is secured.

B. Session Key Generation using AVISPA

Session key is generated between DCH and BS with the help of RCH. The session key is generated using the random numbers generated by DCH and RCH. During communication, BS checks the authenticity of DCH using track and secor numbers of DCH and RCH. Fig. 15 shows the roles specified by DCH and RCH. Fig. 16 shows the role of BS and session, environment and goal sections of HLPSL code. Fig. 17 shows the simulation results of Session key generation by OFMC model and CL-AtSe model.

```
% Session Key Generation: Role of Data Cluster Head (DCH)
role dch(D,R:agent,
         Hsh : hash_func,
         KDR,KSD:symmetric_key,
         TND,SND,PSWd  :text,
       Snd,Rcv  :channel(dy))
played_by D
def=
local State                :nat,
ND,NX,AIDD,V1,V4T                  :text,
NRT,V4,TSDRT,NR,TSDR,VRT,KDRT :text,
SK : symmetric_key
const sec_K_DR             :protocol_id
init State :=0
transition
1. State=0          /\ Rcv(start)  =|>
  State':=1         /\ ND' := new() /\ NX' := xor(KDR,ND')
                    /\ AIDD' := Hsh(D.KDR.ND'.TND.SND)
                    /\ V1' := Hsh(AIDD'.KDR.NX')
                    /\ Snd(D.NX'.AIDD'.V1'.R)
1. State=1          /\ Rcv(R.NRT'.TSDRT'.V4'.D)  =|>
  State':=2         /\ NR' := xor(Hsh(KDR.D.ND),NRT')
                    /\ TSDR' := xor(Hsh(KDR.D.ND),TSDRT')
                    /\ V4T' := Hsh(NR'.ND.TSDR'.KDR) %
verify
                    /\ SK' := xor(ND.NR')
                    /\ KDR' := Hsh(KDR.PSWd.D)

end role
```

(a)

```
% Session Key Generation: Role of Routing Cluster Head (RCH)

role rch(D,R,B  :agent,
 TND,SND,TNR,SNR,PSWr,PSWd:text,
       Hsh  : hash_func, KRB,KDR:symmetric_key,
   Snd,Rcv              :channel(dy))
played_by R
def=
local State              :nat,
ND,AIDD,V1,NR
          :text,
NX,AIDDT,V1T,T1,NRT,IDDT,V2                  :text,
V3,V3T,T2,KRBT,TSDRT,TSDR,V4,NDT,KDRT        :text
const sec_K_RB,sec_K_RD  :protocol_id
init State :=0
transition
1. State=0  /\  Rcv(D.NX'.AIDD'.V1'.R) =|>
  State':=1 /\ ND' := xor(KDR,NX') /\ AIDDT' := Hsh(D.KDR.ND'.TND.SND)
            /\ V1T' := Hsh(AIDD'.KDR.NX')  /\ NR' := new()
            /\ T1' := new()          /\ NRT' := xor(Hsh(KRB.T1'),NR')
            /\ IDDT' := xor(Hsh(KRB.T1'.NR'),D)
            /\ NDT' := xor(Hsh(D.T1'.KRB),ND')
            /\ V2' := Hsh(D.NR'.ND'.T1'.KRB)
            /\ Snd(R.NRT'.IDDT'.NDT'.T1'.V2'.B)
1. State=1 /\ Rcv(B.T2'.V3'.R) =|>
  State':=2/\ V3T' := Hsh(NR.ND.KRB.D.T2') /\ KRB' := Hsh(KRB.PSWr.R)
            /\ NRT':= xor(Hsh(KDR.D.ND),NR)/\ TSDR' := new()
            /\ TSDRT' := xor(Hsh(KDR.D.ND),TSDR')
            /\ V4' := Hsh(NR.ND.TSDR'.KDR)
            /\ KDR' := Hsh(KDR.PSWd.D)   /\ Snd(R.NRT'.TSDRT'.V4'.D)
end role
```

(b)

Fig. 15. Session key generation: Role specification of in HLPSL (a)Data Cluster Head (b)Routing Cluster Head

```
%Session Key Genearation: Role of Base Station
role basestation(R,B  :agent,
            PSWr      :text,
            Hsh       : hash_func,
            KRB       :symmetric_key,
            Snd,Rcv              :channel(dy))
played_by B
def=
local State:nat,
NR,NRT,IDDT,NDT,V2                  :text,
T1,IDD,ND,V2T,T2,V3,KRBT      :text,
SK:symmetric_key
const sec_K_BR                      :protocol_id
init State :=0
transition
1. State=0          /\ Rcv(R.NRT'.IDDT'.NDT'.T1'.V2'.B)  =|>
  State' :=1        /\ NR' := xor(Hsh(KRB.T1'),NRT')
                    /\ IDD' := xor(Hsh(KRB.T1'.NR'),IDDT')
                    /\ ND' := xor(Hsh(IDD'.T1'.KRB),NDT')
                    /\ V2T' := Hsh(IDD'.NR'.ND'.T1'.KRB)
                    /\ SK' := xor(ND',NR') /\ T2' := new()
                    /\ V3' := Hsh(NR'.ND'.KRB.IDD'.T2')
                    /\ KRB' := Hsh(KRB.PSWr.R)
                    /\ Snd(B.T2'.V3'.R)
end role
```

(a)

```
role session(D,R,B:agent
   TND,SND,TNR,SNR,PSWd,PSWr  :text,
    KDR,KSD,KRB: symmetric_key,
    Hsh       : hash_func)
def=
local SB,RB,SR,RR,SD,RD          : channel(dy)
composition
dch(D,R,Hsh,KDR,KSD,TND,SND,PSWd,SD,RD) /\
rch(D,R,B,TND,SND,TNR,SNR,PSWr,PSWd,Hsh,KRB,KDR,SR,RR) /\
basestation(R,B, PSWr,Hsh,KRB, SB,RB)
end role
%-----------------------------------------------------------------
role environment() def=
          const k_br,k_dr, k_rb, k_rd      :protocol_id,
          d,r,b:agent,
          kdr,ksd,krb :symmetric_key,
          tnd,snd,tnr,snr,pswd,pswr:text,
          hsh  :hash_func
intruder_knowledge = {d,r,b}
composition
session(d,r,b,tnd,snd,tnr,snr,pswd,pswr,kdr,ksd,krb,hsh)
end role
%-----------------------------------------------------------------
goal
secrecy_of sec_K_BR,sec_K_DR,sec_K_RB,sec_K_RD
authentication_on k_br
authentication_on k_rb
authentication_on k_dr
authentication_on k_rd
end goal
%-----------------------------------------------------------------
environment()
```

(b)

Fig. 16. Session key generation: Role specification of in HLPSL (a)Base Station (b)Session, Goal and Environment

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL

C:\progra~1\SPAN\testsuite\results\Session.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.10s
  visitedNodes: 10 nodes
  depth: 3 plies
```

(a)

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  C:\progra~1\SPAN\testsuite\results\Session.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 1 states
  Reachable  : 1 states
  Translation: 0.03 seconds
  Computation: 0.00 seconds
```

(b)

Fig. 17. Simulation results of Session key generation: (a) OFMC model, (b) CL-AtSe model

TABLE II. SYMMETRIC KEY GENERATION: COMPUTATION OVERHEAD AT BS, RCH, DCH AND SENSOR NODE

Message	BS	RCH	DCH	Sensor Node
Message 1	$3T_H + 2T_X$	$3T_H + 2T_X$		
Message 2		$4T_H + 3T_X$	$4T_H + 3T_X$	
Message 3			$4T_H + 3T_X$	$4T_H + 3T_X$
Total	$3T_H + 2T_X$	$7T_H + 5T_X$	$8T_H + 6T_X$	$4T_H + 3T_X$

TABLE III. SESSION KEY GENERATION: COMPUTATION OVERHEAD AT BS, RCH AND DCH

Message	BS	RCH	DCH
Message 1		$2T_H + T_X$	$2T_H + T_X$
Message 2	$4T_H + 3T_X$	$4T_H + 3T_X$	
Message 3	T_H	T_H	
Message 4		$3T_H + 2T_X$	$3T_H + 2T_X$
Session key computation	T_X	T_X	
Key freshness	T_H	$2T_H$	T_H
Total	$6T_H + 4T_X$	$12T_H + 7T_X$	$6T_H + 3T_X$

TABLE IV. DATA TRANSFER PHASE: COMPUTATION OVERHEAD AT BS, RCH, DCH AND SENSOR NODE

Message	BS	RCH	DCH	Sensor Node
Message 1			$T_D + T_H$	$T_E + T_H$
Message 2		T_H	$T_{Agg} + T_E + T_H$	
Message 3	$2T_D + T_H$	$T_E + T_H$		
Total	$2T_D + T_H$	$T_E + 2T_H$	$T_{Agg} + T_E + T_D + 2T_H$	$T_E + T_H$

TABLE V. MAKA: COMPUTATION OVERHEAD

Phase	BS	CH	CM
Initialization	$1T_{SM}$		
Key generation		$1T_{HG} + 1T_{SM}$	$1T_{HG} + 1T_{SM}$
Node Registration	$mT_{AD} + mT_{SM} + n\,(1T_{SM})$	$1T_{AE}$	-
Authentication		$3T_{SM} + 1T_{AE} + 1T_{AD}$	$3T_{SM} + 1T_{AE} + 1T_{AD}$
CM $\leftrightarrow$ CH			
Authentication	$1T_{AD} + 1T_{SM}$		
CH $\leftrightarrow$ BS			
Session key agreement	$1T_P$	-	$1T_P$

VII. PERFORMANCE EVALUATION

The aim of the proposed LSSCW scheme is to provide the security for data transfer in the cluster based WSN and at the same time the scheme should be lightweight in terms of communication and computation overhead. Communication overhead is related to the number of message transfers in the network whereas computation overhead deals with the time required for execution of the scheme.

1) Communication Overhead: In LSSCW, number of message transfers takes place during shared symmetric key generation phase, session key generation phase and data transfer phase. The network contains n sensor nodes, m DCH and m RCH nodes. Symmetric key generation process requires $2m + n$ message transfers and session key generation needs $4m$ messages to generate session keys for each DCH. During data transfer phase, each sensor node periodically send the data to DCH. After aggregation process at DCH, only 2 messages per DCH are required to send the data till BS.
2) Computation Overhead: LSSCW uses hash and exclusive OR operations for symmetric key and session key generation. During the data transfer, the data is encrypted using shared symmetric and session keys. Consider the time required for hash function is T_H, time required for Exclusive OR operation is T_X, Time required for encryption is T_E and time required for decryption is T_D. Time required by DCH for aggregation operation is T_{Agg}. Table II shows the number of operations required at each node during symmetric key generation. Table III shows number of operations required for session key generation. Data transfer phase requirement is defined in Table IV. To define the total computation cost, we neglect Exclusive OR operation as the time requirement for Exclusive OR operation is very very small. Considering the network contains one BS, m RCH, m DCH and n sensor nodes, Computation cost for symmetric key generation phase is $(14m + 8n)T_H$ and session key generation phase is $(20m)T_H$.

A. Comparative Analysis

In this section we presented the comparative analysis of the proposed work with Mutual Authentication and Key Agreement (MAKA) scheme presented by Harbi et al. [38] in the year 2019 and few other authentication and key agreement schemes. For comparison, we have considered the operations related to key generation process of LSSCW. MAKA defines the cluster based network model consisting of BS, CH and Cluster Members (CMs) which are sensor nodes. MAKA has five phases: initialization, key generation, node registration, node authentication, and session key agreement. Following are the notations used for basic operations performed in MAKA.

- T_{HG}: Time for hash function on G
- T_{SM}: Time for ECC scalar multiplication
- T_{PA} : Time for ECC point addition
- T_{AE}: Time for asymmetric encryption
- T_{AD}: Time for asymmetric decryption
- T_P: Time for pairing on G
- m: Number of cluster heads in the network
- n: Number of cluster members in the network

TABLE VI. COMPARISON OF MAKA AND LSSCW

Scenario 1			Scenario 2		
Sensor nodes	MAKA	LSSCW	CH	MAKA	LSSCW
10	473.735	0.2622	1	473.735	0.2622
20	902.445	0.4462	2	947.47	0.5244
30	1331.155	0.6302	3	1421.205	0.7866
40	1759.865	0.8142	4	1894.94	1.0488
50	2188.575	0.9982	5	2368.675	1.311
60	2617.285	1.1822	6	2842.41	1.5732
70	3045.995	1.3662	7	3316.145	1.8354
80	3474.705	1.5502	8	3789.88	2.0976
90	3903.415	1.7342	9	4263.615	2.3598
100	4332.125	1.9182	10	4737.35	2.622

We have computed the number of operations performed in each phase. Table V shows the number of operations performed for MAKA scheme. During initialization phase BS generates the public key and sends to all other nodes. As the part of key generation phase, (public key, private key) pairs are generated for each CMs. Node registration phase is responsible for registering CHs and CMs at BS end. In the table, we have defined total cost for node registration considering m CHs and n CMs. Authentication phase is designed for mutual authentication between CM and CH, CH and BS. Finally, session key is generated at BS and CM end.

Considering the network contains one BS, m CHs and n CMs, the computation cost related to BS is $2m\,(T_{AD} + T_{SM}) + n\,(T_{SM} + T_P)$, total computation cost related to m CHs is $m\,(T_{HG} + 4T_{SM} + 2T_{AE} + T_{AD})$ and the cost related to sensor n nodes is $n\,(T_{HG} + 4T_{SM} + T_{AE} + T_{AD} + T_P)$. Thus the total cost for the MAKA scheme is $(m+n)T_{HG} + (6m+5n)T_{SM} + (2m+n)T_{AE} + (3m+n)T_{AD} + 2nT_P$.

To compare the computation time between MAKA and LSSCW, we exploited the time requirement for each basic operation presented in Kilinc and Yanik [39]. Thus, time required for one-way hash function T_H = 0.0023 ms, symmetric key encryption / decryption = T_E = T_D = 0.0046 ms, hash function on group G T_{HG} = 12.419 ms, ECC scalar multiplication T_{SM} = 2.226 ms, ECC point addition T_{PA} = 0.0288 ms, asymmetric encryption/decryption T_{AE} = T_{AD} = 3.85 ms, pairing on G T_P = 5.811 ms and MAC T_{MAC} = 0.0046 ms.

Two scenarios are considered for comparison between MAKA and LSSCW. In the first scenario, the network contains one BS, one cluster and number of sensor nodes varies from 10 to 100. In the second scenario, the network contains one BS, number of clusters vary from 1 to 10 where each cluster contains 10 sensor nodes each. Table VI shows the computation time of Scenario 1 and Scenario 2 for LSSCW and MAKA. The results in the Table VI indicates that LSSCW is very lightweight compared to MAKA as the average computation overhead is lesser by approximately 30%.

Fig. 18 shows the logarithmic graph of comparative computation cost of LSSCW and other authentication and key agreement schemes. Table VII summarises the computation costs

TABLE VII. COMPUTATION COST COMPARISON

Sr. No.	Scheme	Computation cost	Time (in ms)
1	LSSCW	$22T_H + 24T_H$	0.1058
2	[38]	$6T_{SM} + 5T_{AE/AD} + 3T_P$	50.039
3	[40]	$3T_{HG} + 7T_{SM} + 2T_{PA} + 2T_P$	64.5186
4	[41]	$10T_H$	0.023
5	[42]	$18T_H$	0.0414
6	[16]	$22T_H$	0.0506
7	[17]	$26T_H$	0.0598
8	[43]	$19T_H + 2T_{SM}$	4.4957
9	[44]	$10T_H + 2T_{SM}$	4.475
10	[45]	$17T_H + 2T_{SM}$	4.4911
11	[35]	$2T_{HG} + 21T_{SM} + 11T_{PA}$ $+2T_E + 4T_H + 6T_{MAC}$	71.9468

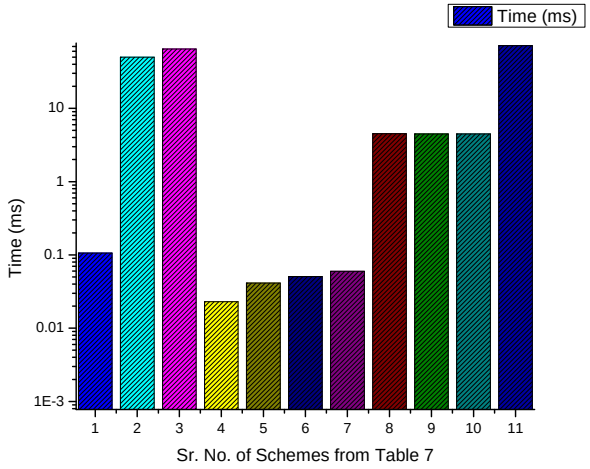

Fig. 18. Computation cost comparison

related to the LSSCW scheme and other schemes presented in [16], [17], [35], [38], [40]–[45]. LSSCW is more efficient than [35], [38], [40], [43]–[45] in terms of time requirement. The LSSCW performance is 0.1058 ms which is far better than the performance of [44] which is 4.475 ms.

Compared to [16], [17], [41], [42], LSSCW requires few additional hash operations for session key generation process. The session key is used in the data transfer phase for secure transfer and verification of data between DCH and BS. Hence, even though LSSCW requires few additional hash operations, it provides end-to-end security to the transferred data. LSSCW session key helps in sender authentication and protection against Man-in-the-middle attack. The schemes presented in [16], [17], [41], [42] do not use session key thereby providing a less secure environment compared to LSSCW for data communication in WSN.

VIII. CONCLUSION

Wireless sensor network needs solutions for securing data during communication while handling large number of sensor

nodes. In this paper, we have proposed a security framework which covers details of cluster management, key generation and secured data transfer. The lightweight solution for security based on hash and exor makes the proposed work suitable for WSN. Security analysis is done using AVISPA tool which proves that the proposed work is secured and is efficient in terms of time requirement.

The analysis shows that LSSCW is approximately 30% better computation time requirement compared to MAKA. Though the computation time for LSSCW is greater than few protocols in the literature, its session key mechanism provides security against MITM attack. For the session key based protocols in the literature, it is observed that the LSSCW scheme offers a better performance in terms of computation cost.

As the part of future work, we expect to evaluate the performance of the proposed scheme on actual hardware devices. Based on the experimentation, we will examine the effectiveness of the scheme for various applications.

REFERENCES

[1] K. Sohraby, D. Minoli, and T. Znati, "Wireless sensor networks: technology, protocols, and applications," *John Wiley & Sons*, pp. 203–209. ISBN 978-0-471-74300-2.

[2] W. Dargie, and C. Poellabauer, "Fundamentals of wireless sensor networks: theory and practice," *John Wiley & Sons*, pp. 168–183, 191–192, 2010, ISBN 978-0-470-99765-9.

[3] M. S. Karyakarte, A. S. Tavildar, and R. Khanna, "Dynamic node deployment and cross layer opportunistic robust routing for PoI coverage using WSNs," *Wireless Personal Communications*, Vol. 96, No. 2, 2741-2759,2017.

[4] C. Wei, J. Yang, Y. Gao, and Z. Zhang, "Cluster-based routing protocols in wireless sensor networks: A survey," *In Proceedings of 2011 International Conference on Computer Science and Network Technology*, Vol. 3, pp. 1659–1663, December 2011, IEEE.

[5] C. F. García-Hernández, P. H. Ibarguengoytia-Gonzalez, J. García-Hernández, and J. A. Pérez-Díaz, "Wireless sensor networks and applications: a survey," *IJCSNS International Journal of Computer Science and Network Security*, Vol. 7, No. 3, pp. 264-273, 2007.

[6] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, Vol. 54, No. 15, pp. 2787–2805, 2010

[7] A. S. K. Pathan, H. W. Lee, and C. S. Hong,"Security in wireless sensor networks: issues and challenges," *In 2006 8th International Conference Advanced Communication Technology*, Vol. 2, pp. 6-pp, IEEE, 2006

[8] N. Gautam, W. I. Lee, and J. Y. Pyun, "Track-sector clustering for energy efficient routing in wireless sensor networks," *In 2009 Ninth IEEE international conference on computer and information technology*, Vol. 2, pp. 116–121, IEEE, October 2009.

[9] S. M. Jung, Y. J. Han, and T. M. Chung, "The concentric clustering scheme for efficient energy consumption in the PEGASIS," *In The 9th international conference on advanced communication technology*, Vol. 1, pp. 260–265, IEEE, February 2007.

[10] R.V.Saraswathi, L. P. Sree, and K. Anuradha, "Multi-stage Key Management Scheme for Cluster based WSN,", *International Journal of Communication Networks and Information Security*, Vol. 10, No. 3, pp. 552, 2018.

[11] X. Yi, R. Paulet, and E. Bertino, "Homomorphic encryption and applications," *Springer Briefs in Computer Science*, Vol. 3, Cham: Springer, 2014.

[12] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks," *IEEE Access*, Vol. 5, pp.3376–3392, 2017.

[13] S. Athmani,A. Bilami, and D. E. Boubiche,"EDAK: An efficient dynamic authentication and key management mechanism for heterogeneous WSNs," *Future Generation Computer Systems*, Vol. 92, pp.789–799, 2019.

[14] M. Turkanovic, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks based on the Internet of Things notion," *Ad Hoc Networks*, Vol. 20, pp. 96–112, 2014.

[15] Q. Chang, Y. P. Zhang, and L. L. Qin, "A node authentication protocol based on ECC in WSN," *In 2010 International Conference On Computer Design and Applications*, Vol. 2, pp. V2-606, IEEE, 2010.

[16] M. S. Farash, M. Turkanović, S. Kumari,and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Networks*, 36 (2016), pp. 152–176, 2016.

[17] R. Amin, S. H. Islam, G. P. Biswas, and M. S. Obaidat,"A robust mutual authentication protocol for WSN with multiple base-stations," *Ad Hoc Networks*, Vol. 75, pp. 1–18, 2018.

[18] M. Vaneeta,and S. S. Kumar,"NPKG: Novel Pairwise Key Generation for Resisting Key-based Threats in Wireless Sensor Network," *IJ Network Security*, Vol. 21, No. 1, pp.122–129, 2019.

[19] B. Hu,and H. Gharavi, "Smart grid mesh network security using dynamic key distribution with merkle tree 4-way handshaking," *IEEE Transactions on Smart Grid*, Vol. 5, No. 2, pp.550–558,2013.

[20] S. J. Choi, K. T. Kim, and H. Y. Youn,"An energy-efficient key predistribution scheme for secure wireless sensor networks using eigenvector," *International Journal of Distributed Sensor Networks*, Vol. 9, No. 6, pp.216754, 2013.

[21] S. Bag,and B. Roy,"A new key predistribution scheme for general and grid-group deployment of wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, Vol. 2013, No.1, pp. 145, 2013.

[22] W. Bechkit, Y. Challal, A. Bouabdallah, and V. Tarokh, "A highly scalable key pre-distribution scheme for wireless sensor networks," *IEEE Transactions on Wireless Communications*, Vol. 12, No.2, pp. 948–959, 2013.

[23] G. R. Pathak, and S. H. Patil,"Mathematical Model of Security Framework for Routing Layer Protocol in Wireless Sensor Networks," *Procedia Computer Science*, Vol. 78, pp. 579–586, 2016.

[24] H. L. Yeh, T. H. Chen, P.C. Liu, T. H. Kim, and H. W. Wei,"A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, Vol. 11, No.5, pp. 4767–4779, 2011.

[25] X. Huang, P. Shah, and D. Sharma,"Fast algorithm in ECC for wireless sensor network," *In Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 2, pp. 17–19, 2010.

[26] O. Arazi, and H. Qi, "Self-certified group key generation for ad hoc clusters in wireless sensor networks," *In Proceedings. 14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005*, pp. 359-364, IEEE, October 2005.

[27] O. Arazi, I. Elhanany, D. Rose,H. Qi, and B. Arazi,"Self-certified public key generation on the intel mote 2 sensor network platform," *In 2006 2nd IEEE Workshop on Wireless Mesh Networks*, pp. 118–120. IEEE, September, 2006.

[28] S. U. Khan, C. Pastrone,L. Lavagno, and M. A. Spirito,"An authentication and key establishment scheme for the IP-based wireless sensor networks," *Procedia Computer Science*, Vol. 10, pp. 1039–1045, 2012.

[29] Z. Qin, X. Zhang, K. Feng, Q. Zhang,and J. Huang, "An efficient key management scheme based on ECC and AVL tree for large scale wireless sensor networks," *International Journal of Distributed Sensor Networks*, Vol. 11, No. 9, pp. 691498, 2015.

[30] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *International Conference on the Theory and Applications of Cryptographic Techniques*, Berlin, Heidelberg, pp.223–238, 1999.

[31] I. Nadir, W. K. Zegeye, F. Moazzami, and Y. Astatke, "An efficient key management scheme based on ECC and AVL tree for large scale wireless sensor networks," emphInternational Journal of Distributed Sensor Networks, Vol. 11, No. 9, pp. 691498, October 2016.

[32] A. A. Agarkar, and H. Agrawal, "J-PAKE and ECC Based Authentication Protocol for Smart Grid Network", *In International Conference on Advances in Computing and Data Sciences*, pp. 507–522. Springer, Singapore, April 2018.

[33] A. A. Agarkar, and H. Agrawal,"Lightweight R-LWE-based privacy preservation scheme for smart grid network," *International Journal of Information and Computer Security*, Vol. 11, No. 3, pp. 233–254, 2019.

[34] A. A. Agarkar, and H. Agrawal, "LRSPPP: lightweight R-LWE-based secure and privacy-preserving scheme for prosumer side network in smart grid," *Heliyon*, Vol. 5, No. 3, e01321, 2019.

[35] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future generation computer systems*, Vol. 78, pp. 956–963, 2018.

[36] El-hajj, M., Fadlallah, A., Chamoun, M., and Serhrouchni, A., "A survey of internet of things (IoT) Authentication schemes," *Sensors*, Vol. 19, No. 5, 1141, 2019.

[37] Al-Zubaidie, M., Zhang, Z., and Zhang, J., "Ramhu: A new robust lightweight scheme for mutual users authentication in healthcare applications," *Security and Communication Networks*, 2019.

[38] Y. Harbi, Z. Aliouat, A. Refoufi, S. Harous, and A. Bentaleb, "Enhanced Authentication and Key Management Scheme for Securing Data Transmission in the Internet of Things,", *Ad Hoc Networks*, pp. 101948, 2019.

[39] H.H. Kilinc, and T. Yanik, "A survey of SIP authentication and key agreement schemes," *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 2, pp. 1005–1023, 2013.

[40] A. Mehmood, M. M. Umar, and H. Song, "ICMDS: Secure inter-cluster multiple-key distribution scheme for wireless sensor networks," *Ad Hoc Networks*, Vol. 55, pp. 97–106, 2017.

[41] A. Gupta, M. Tripathi, T. J. Shaikh, and A. Sharma, "A lightweight anonymous user authentication and key establishment scheme for wearable devices," *Computer Networks*, Vol. 149, pp. 29–42, 2019.

[42] D. Mishra, P. Vijayakumar, V. Sureshkumar, R. Amin, S. H. Islam, and P. Gope, "Efficient authentication protocol for secure multimedia communications in IoT-enabled wireless sensor networks," *Multimedia Tools and Applications*, Vol. 77, No. 14, pp. 18295–18325, 2018.

[43] C. Wang, G. Xu, and J. Sun, "An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks," *Sensors*, Vol. 17, No. 12, pp. 2946, 2017.

[44] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah,and S. Kumari, "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 8, pp. 3599–3609, 2017.

[45] F. Wu, L. Xu, S. Kumari, and X. Li, "A privacy-preserving and provable user authentication scheme for wireless sensor networks based on internet of things security," *Journal of Ambient Intelligence and Humanized Computing*, Vol. 8, No. 1, pp. 101–116, 2017.