IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 February 2023, accepted 6 March 2023, date of publication 10 March 2023, date of current version 24 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3255646

RESEARCH ARTICLE

EN-LAKP: Lightweight Authentication and Key Agreement Protocol for Emerging Networks

NEMALIKANTI ANAND, (Graduate Student Member, IEEE), AND M. A. SAIFULLA, (Member, IEEE)

School of Computer and Information Sciences, University of Hyderabad, Hyderabad, Telangana 500046, India

Corresponding author: M. A. Saifulla (saifullah@uohyd.ac.in)

ABSTRACT In the next generation, emerging network technologies like Software Defined Networking (SDN) and Wireless Sensor Networks (WSNs) will be developed and deployed to improve computing facilities. Micro-electromechanical system (MEMS) technologies advance, and WSNs gain in popularity because they provide real-time monitoring solutions that are both economically and practically viable. The SDN paradigm is consequently being incorporated into WSN to prevent a performance bottleneck with traditional network architecture as network traffic and diverse sensor nodes increase. Because they interact through a public channel and the sensor nodes are spread throughout a hostile environment, the information transmitted among entities is subject to assault. The proposed protocol showed how centralized SDN controller nodes logically assumed the role of network management and control. Therefore, we present a Lightweight Authentication and Key Agreement Protocol (LAKP) for SDN-enabled WSNs to protect entity communication. Additionally, we demonstrate that the suggested system prevents known security flaws by conducting both informal and formal security studies using the Scyther tool and Burrows-Abadi-Needham (BAN) logic. Further, the performance study demonstrates that the suggested scheme performs better in computing and communication burdens than related protocols with 1.6% to 5.4% and 1.3% to 5.8%, respectively.

INDEX TERMS Software defined networks, wireless sensor networks, authentication, security, key agreement, Scyther.

I. INTRODUCTION

With the advancement of wireless network and MEMS technology, WSNs are becoming more and more common because they offer solutions for real-time monitoring that are both reasonably priced and practically viable. WSNs are currently often employed in unattended contexts, in addition to a variety of real-time applications such as ''traffic monitoring,'' ''environmental control,'' ''military surveillance,'' ''vehicle tracking,'' ''healthcare monitoring,'' ''habitat monitoring,'' and ''animal monitoring''. Wireless sensors in hazardous areas would be quick and simple to setup. Furthermore, WSNs are becoming sophisticated and an intrinsic part of people's daily life, according to [1]. External users in many important applications are often interested in getting real-time information from sensor nodes [2], [3]. Fast-evolving Internet of Things (IOT) has given rise to intelligent systems like smart homes, e-healthcare, wearable technology, and electric vehicle charging stations [4], [5], [6]. According to [7], more than 85 percent of firms will leverage IoT devices such as sensor nodes in various ways, and over 90 percent of these enterprises are unsure about smart device security.

WSNs are made up of many low-power, low-cost sensor units that may be distributed across a large region. As previously said, WSNs have arisen as a popular and rising technology in a multitude of industries. One of their most significant challenges is providing a secure connection through their wireless channel. Due to their limited resources, traditional security methods cannot be employed because they use much energy. As a result, WSNs recently drew the attention of academics, who are now focusing on WSN security [8].

The quantity of connected devices is exponentially increasing in the current day. These clever gadgets are used to carry out ''Distributed Denial of Service (DDoS) attacks'' and the

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

spread of malware. Because of its weak built-in protection and distinctive architecture, it will always be the top pick for cybercriminals [9], [10]. Furthermore, IoT industry giants are producing intelligent goods to conquer the open market, which means products must be made without regard for security. Due to this paradox and extra resource restrictions on IoT devices, traditional host-based safeguard solutions such as antivirus, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS) have failed to defend low-power intelligent devices. Hence, a prominent security system that is flexible and proactive is needed to protect users, networks, data, and devices. Yu et al. [11] proposed a security framework for networked environments by leveraging SDN to circumvent the restricted resources of intelligent devices.

SDN is a new networking paradigm that allows network components to be programmed. This networking architecture enables the research community to put their theories about building and testing new protocols to the test using application software. These are typically put to the test by being deployed on the controller. A centralized design is inherited by SDN where the central entity in charge of switching and routing functionalities is the controller. Apart from that, connected hosts and switches in the data plane should communicate with the controller via a southbound interface [12]. In general, the typical architecture of SDN environment should comprise three layers, such as the application layer, the control layer, and the data layer. Network policies can be defined at the application plane in bespoke apps. The customized apps are stored on the application plane and can communicate with the controller via the Representational State Transfer (REST) application programming interface. A southbound Application Programming Interface (API) connects the control plane to the data plane. Both the control plane and the data plane are isolated from each other. Similarly, network functionality is logically centralized, enabling the controller to respond reactively as well as proactively like add, delete, and change flow elements (predefined rules). SDN also provides rapid attack response, ''granular traffic filtering,'' and ''dynamic security rule deployment''. SDN depends heavily on the OpenFlow to define network device security policy rules. The controller promotes a global network view by keeping a link via the OpenFlow switches [13], [14]. Because third-party apps are used in the design, the network's various planes may be vulnerable to various attacks.

Lastly, an SDN controller may enable computing and analyzing the security of every authentication request at the network level. The controller node should secure a simple authentication module, as this is best done at a low level entity. The privatization standards are further improved by ensuring that SDN features protect users' personal and private information.

A. RELATED WORK

Several protocols have been suggested in the current literature to fulfill the privacy and security needs of WSN. Several researchers recently presented authentication protocols for WSN environments. Zhang et al. [15] created a novel authentication technique that protects user privacy and employs only lightweight cryptographic basics. However, ensuring user anonymity has failed to be secured. Two authentication techniques have been proposed by Chang and Le [16] firstly, the technique employing bitwise XOR and hash operations, and secondly, on the contrary, the employment of an extra Elliptic Curve Cryptography (ECC) operation. Evidently, their initial rudimentary method later turned out to be vulnerable against a session key breach attack. Their schemes, however, are vulnerable to ''session-specific information leaks'' and ''offline password guessing attacks,'' according to [17].

Gope and Hwang [18] proposed an authentication technique to transport real-time data in a WSN environment. It uses symmetric encryption and decryption, user transactions, and pseudonym key information to address the required qualities. However, their strategy asks for Gateway Node (GW) to save additional user data, which is impractical to achieve and results in failed authentication. Similarly, this protocol is vulnerable to Ephemeral Secret Leakage (ESL) and cloning attacks, as well as desynchronization issues. Furthermore, because SN's identity is disclosed while sending over a public channel, their approach does not allow anonymity or untraceability. As a result, their approach is subject to node capture attempts [19]. Other side, Their scheme provides forward and backward secrecy property for session key. However, they have not addressed the formal security analysis of the proposed protocol.

Li et al. [20] described a three-factor authentication scheme for industrial IoT. ECC is used to keep transmitted communications anonymous, unlinkable, and fresh [21]. In fact, the messages are transmitted without timestamps, which more helpful to verify the freshness of message. In case, an adversary replays any message, he or she will successfully complete verification process. Cloning and ESL attacks can also be employed against the aforementioned protocol.

Li et al. [22] proposed ECC-based three-factor authentication mechanism for an IoT environment. The long-term secret is shared between GW and SN during the deployment phase. When an adversary seizes the SN and attempts to retrieve the long-term key which kept in memory of SN, the SN's previous session keys are easily discovered. Hence, their scheme is prone to ''replay attacks,'' ''clone attacks,'' and does not support forward secrecy.

A lightweight and physically secure anonymous mutual authentication protocol for the industrial WSN environment was discussed in Gope et al. [23]. The authors adopted primitive ''Physically Unclonable Functions (PUFs)'' [24] to enable physical security for Mobile User (MU) and SN. In addition, the GW would have to verify a large set of Challenge-Response Pair (CRP) with regard to SN and users, which proved difficult. Their scheme is vulnerable to insider attack and ESL attack as well. Further, PUF based protocols are vulnerable to machine learning attacks on the other side.

Banerjee et al. [25] discusses a dynamic pseudonym-based authentication scheme. Because the authentication stage assumes a direct relationship between SN and MU, fewer message transfers are necessary. These protocols, according to [22], are unsuitable for IoT contexts. To guarantee anonymity and unlinkable communication, MU in [25] changes his or her alias after each session. However, a desynchronization attack exposes this functionality to privacy concerns. Furthermore, various cloning malpractices and forward secrecy may affect the protocol. They used the widely established Real-Or-Random (ROR) model, BAN logic, and an Automated Validation of Internet Security Protocols and Applications (AVISPA) software simulation tool to show the security of their system.

Iqbal et al. [26] devised a lightweight authentication scheme for SDN-based smart homes. To defend against both passive and active threats, it leverages symmetric key encryption and decryption. ''Mutual Authentication and Key Agreement (MAKA)'' between MU and SN is left unaddressed in the protocol. Because no credentials are required, any user may begin a session on the device. Cloning and device-specific registration attacks are possible using their approach. It becomes increasingly computationally costly as the number of users grows. They conducted an informal and formal study of the suggested method using BAN logic and the ProVerif tool to demonstrate how their protocol archives good security. Furthermore, they stated that their protocol is appropriate for devices with limited resources.

Sutrala et al. [27] suggested a MAKA protocol for industrial cyber-physical systems based on SDN. ECC is used by users to address the relevant characteristics. To provide anonymous and unlinkable communication, MU connects with Software Defined Networking Controller (SDNC) throughout the authentication process and updates it after the connection ends. This configuration, however, exposes the protocol to tracing and desynchronization attacks. The aforementioned protocol has high computing, communication, and storage overheads, as well as being subject to cloning attacks, making it unsuitable for WSN systems with restricted resources. They claimed that their protocol ensures security services and proved protocol's security with AVISPA simulation tool. They adopted Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL) library based testbeds to obtain computational times of cryptographic primitives.

Roy and Bhattacharya [28] developed an authentication protocol for industrial wireless sensor network to enable real time data transfer among entities. Their scheme employed ECC due to its low computational time. They proposed a velocity based approach to mitigate cloning attacks as well as counter various security attacks. Their protocol has been proven both formally and informally.

Recently, Li et al. [29] developed a provably secure authentication technique for a 5G-enabled WSN environment using one-way hash functions and XOR operations. Initially, they performed cryptanalysis on Yu and Park [30]'s protocol and demonstrated that it is vulnerable to sensor capture and temporary information disclosure attacks. They then presented an improved protocol for a 5G-enabled WSN environment. They assessed the proposed protocol using both formal (ROR Model, BAN logic, and ProVerif) and informal security analyses. Furthermore, they stated that their protocol was better to comparable protocols in terms of communication and computing overhead.

B. MOTIVATION AND CONTRIBUTION

According to a review of the literature, most of the existing protocols either fall behind the required security goals or make performance sacrifices. As a result, we propose a LAKP for SDN-enabled WSNs based on hash functions, XOR operations, and concatenation operations. This paper's main contributions are listed below:

- A LAKP leveraging hash functions is proposed for SDN-enabled WSNs.
- The suggested protocol includes innovative features such as dynamic SN addition and a phase for updating user credentials.
- To illustrate its potential to prevent well-known security vulnerabilities, the proposed protocol is carefully evaluated via informal analysis and formal analysis using the Scyther tool and BAN logic.
- According to the usual performance analysis, the suggested protocol is efficient than existing ones with respect to computational and communication performance.

C. ORGANIZATION OF THE PAPER

The remainder of this article is broken into the following sections: In Section II, the pertinent literature is reviewed, with an emphasis on functional, security, and performance features. The network model and the roles of the involved entities are described in Section III. In Section IV, the adversarial paradigm and adversary capabilities are examined. We present our recommended protocol suite in Section V. Both informal and formal security analysis are covered in Section VI. Section VII includes the performance evaluation. Section VIII brings the article to a conclusion.

II. NETWORK MODEL

Figure 1 depicts the proposed network model. It consists of logical control nodes, users, and sensor nodes. The proposed network model maintains one more special module named Registration Authority (RA). The RA is responsible for registration of Users, Sensors, and Control nodes. The RA also stores important parameters submitted by all the registered entities in its local database for further purpose. Similarly, Controller Node (CN) validates the identity of User's and SN with the help of database. The CN accomplishes requests obtained from the application layer and aids in the establishment of session key between user and SN. During the registration process, each user registers with RA and obtains a smart card. For the ones who wish to access the data from

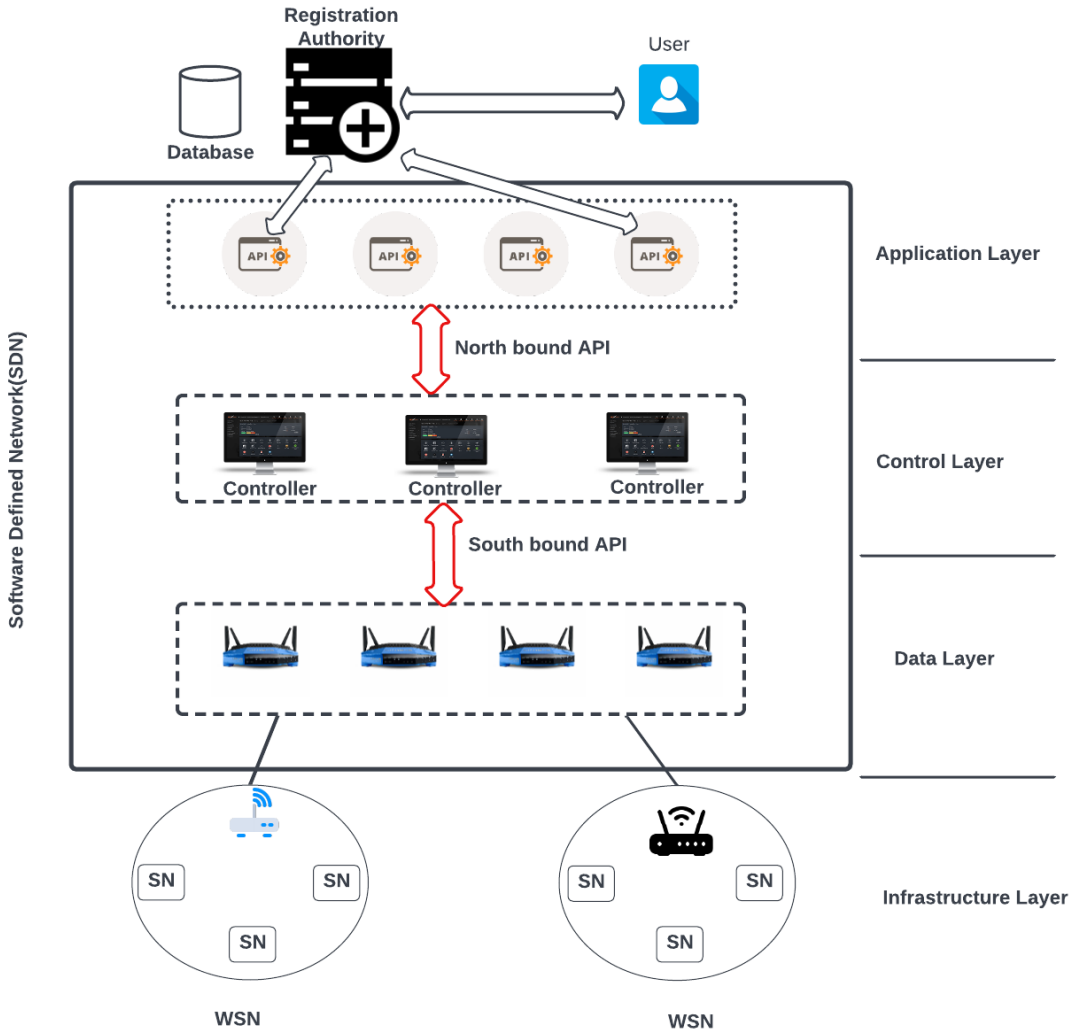

FIGURE 1. Proposed network model.

SN should initiate the login process from the user's device. After a successful login both the user and the SN authenticate one another upon a session key. In sequel, both User and SN use the computed session key to exchange information securely. Furthermore, the interfaces play an important role in providing communication between entities available in various layers. The north bound interface provides an interface between the application layer and controller nodes located at the control layer. Similarly, the southbound interface provides connection between data layer and sensor nodes deployed at the infrastructure layer.

Furthermore, any user who wishes to access RA-deployed SNs in a hostile environment must be registered with RA. Both registered user and SN authenticate with each other through CN and compute a session key. Furthermore, any user who wish to access RA-deployed SNs in hostile environment must be registered with RA. Prior to information exchange, registered user and SN mutually authenticate with each other using CN to establish a session key. Furthermore, we assume that the RA's database is completely secure and unavailable to any opponent. Besides, we hypothesized that CN would have access to the database held by RA. We believed that the RA is in charge of deploying SNs in hostile environments and monitoring their operation. Furthermore, we used the Cipher Block Chaining (CBC) mode of Advanced Encryption Standard (AES) in our protocol for encryption and decryption [31].

III. ADVERSARY MODEL

The primary objective of an adversary is to acquire complete access to network resources. Additionally, in an effort to obtain access to confidential information, they might try to intercept communications that are sent over a channel that participants share [32]. An adversary model is essentially a construct of an attacker, who could be an algorithm or a list of consequences based on skills and aspirations [33]. In this article, the prominent adversary models, i.e., the "Dolev-Yao model" [34], "CK-model" [35] are

adopted, where the adversary can listen to all traffic on a network.

A. SECURITY GOALS

The proposed authenticated key agreement protocol for emerging networks in the presence of an adversary $\mathcal{A}$ should meet the following security goals:

- *Mutual authentication* allows two parties who are communicating to verify their identities.
- *Explicit key authentication* ensures that communicating parties are the only ones who have the correct session key.
- *Key confirmation* occurs when one party guarantees the possession of a session key to another party.
- *Forward secrecy* ensures that even the long-term secret keys revealed, it is impossible to session key by an adversary $\mathcal{A}$ [36].
- *Known key secrecy* is guaranteed if the communicating parties agree upon a session key in a particular session and leakage of past session key do not disclose the current session key [37], [38].
- *Session key freshness* guarantees that the current session key is generated at random and with no regard for previous session keys [37].
- *Unlinkability* of a vehicle is accomplished when an adversary is unable to link different pseudonyms employed by a Vehicle while examining ongoing communication and roaming over several Road Side Units (RSUs) [39].
- *User anonymity* was accomplished for Vehicles when the Trusted Authority (TA) and RSUs were only permitted to access pseudonyms provided by Vehicles [40].
- *Location privacy* of Vehicles achieved if the adversary $\mathcal{A}$ is unable to correlate charging patterns when Vehicles roam across various charging stations employing multiple pseudonyms [40].

IV. THE PROPOSED PROTOCOL SUITE

The proposed protocol comprises the phases such as: *A)* "System Initialization Phase," *B)* Controller Node Registration Phase, *C)* "Sensor Node Registration Phase," *D)* "User Registration Phase," *E)* "Login Phase and Mutual Authentication Phase," *F)* User credential Update Phase, and *G)* Dynamic Sensor Node addition Phase. The notations used in the proposed protocol are shown in Table 2.

A. SYSTEM INITIALIZATION PHASE

The following steps describe how the *RA* generates required parameters in the system:

Step 1: *RA* chooses "collision-resistant one-way cryptographic hash function" $H(.)$.

Step 2: *RA* picks long-term secret keys MSK_{SN_i}, MSK_J, MSK_Q.

TABLE 1. SN registration phase.

SN
1) Generates an identity ID_{SN_i}
2) Forwards a registration request ID_{SN_i}
RA
3) calculates a parameters $K_{SN_i} = H(ID_{ra} \parallel ID_{SN_i} \parallel MSK_{SN_i})$
4) Stores ID_{SN_i}, K_{SN_i} in a table T_{SN}
5) Preloads SN with $< ID_{SN_i}, K_{SN_i} >$ before deploying them in unattended environment

TABLE 2. Notations.

Symbol	Description
ID_u	Identity of User
PW_u	Password of User
ID_{CN_i}	Identity of i^{th} CN
SK_{CN_i}	Long-term secret key of i^{th} CN
R_u, n_1, n_2, n_3	Random numbers
$H(.)$	One-way hash function
MSK_{SN_i}, MSK_J, MSK_Q	Long term secret keys
$\parallel$	Concatenation
$\oplus$	Exclusive-OR operation
AE	Adversary
SK	Session key
RA	Registration Authority
CN	Controller Node
SN	Sensor Node
U	User
SC	Smart Card

Step 3: *RA* generates long-term private keys for CN and SN such as SK_{CN_i}, SK_{SN_i}, respectively, and keeps them secret.

B. SENSOR NODE REGISTRATION PHASE

The SN_i, which is critical to the SDN enabled wireless sensor network environment, can register with the *RA* via secure channel. The *SN* registration procedure with *RA* is depicted below. The *SN* delivers an ID_{SN_i} to RA (Step 1 and 2). *RA* computes K_{SN_i} using MSK_{SN_i} (Step 3). *RA* stores ID_{SN_i}, K_{SN_i}, in local database (Step 4). *RA* Preloads *SN* with $< ID_{SN_i}, K_{SN_i} >$ before deploying them in unattended environment. Table 1 depicts the steps involved in this phase.

C. USER REGISTRATION PHASE

In this phase, the registration facility for U_is provided through secure channel, who wants to become legal user and access the data from sensors through secure channel. Hence, the user starts registration process with the *RA* and procedure has been depicted as below. Initially, user picks a password and identity, then after chooses random number $R_u \in Z_p^*$ (Step 1). Then, after user computes PID_u, PWD_u, Y_u and forwards PID_u, PWD_u to *RA* (Step 2 and 3). Now, the *RA* computes C_u by choosing random number and identity for it and computes N_i (Step 4 and 5). Finally,*RA* sends $< C_u, N_i >$ securely through smart card (Step 6). Other side, User computes parameters V_i, N_i^* (Step 7) and keeps Y_u, V_1 as weel as replaces N_i with N_i^* in smart card (Step 8). At the end, smart

TABLE 3. User registration phase.

User
1) Chooses an identifier ID_u and password PW_u, and a random number $R_u \in Z_p^*$.
2) Then, calculates $PID_u = H(ID_u \parallel R_u)$, $PWD_u = H(pw_u \parallel R_u, Y_u = R_u \oplus H(ID_u \parallel PW_u)$
3) Sends $< PID_u, PWD_u >$
RA
4) Picks a random number r_a and picks identity ID_{ra}, computes $C_u = E_{MSK_J}(r_a \parallel PID_u \parallel ID_{ra})$
5)$N_i = H(PID_u \parallel ID_{ra} \parallel K) \oplus PWD_u$
6) RA delivers $< C_u, N_i >$ securely through smart card
User
7) computes $V_i = (PID_u \parallel PWD_u \parallel R_u)$ and $N_i^* = N_i \oplus H(R_u \parallel PWD_u)$
8) Stores Y_u, V_1 and replaces N_i with N_i^* in smart card
9) Smart card holds parameters $< V_i, C_u, N_i^*, Y_u >$

card holds parameters $< V_i, C_u, N_i^*, Y_u >$ (Step 9). Table 3 depicts the user registration process with *RA*.

D. LOGIN AND MUTUAL AUTHENTICATION PHASE

Once the sensor nodes are deployed, the user U needs to log in the SDN enabled WSN environment through Control Node to access the information available in particular SN_i. Hence, U could launch a login process through ID_u, PW_u. Consequently, both user U and Sensor node SN_i mutually authenticate each other via CN_i and agree upon session key for their further secure communication. The communication entities in this phase are U, SN, and CN. The *User* could launch the login process through ID_u, PW_u (Step 1). The *User* computes $R_u = Y_u \oplus H(ID_u \parallel PW_u)$, $PID_u = H(ID_u \parallel R_u)$, $PWD_u = H(PW_u \parallel R_u)$, $W_S = X_S \oplus PWD_u$ and verifies $V_i = H(PID_u \parallel PWD_u \parallel R_u)$ holds or not. If not holds, SC terminates session (Steps 1-3). The User computes $N_i^! = N_i^* \oplus H(R_u \parallel PWD_u)$, and $D_i = E_{N_i^!}[H(PID_u \parallel C_u \parallel ID_{SN_i} \parallel n_1 \parallel t_1) \parallel ID_{SN_i} \parallel n_1]$ by using random number $n_1 \in Z_p^*$ (Steps 4 and 5). Then user sends the login request message $< M_1$ (Step 6). After receiving message $< M_1$, CN verifies timestamp then proceeds further or abort session (Step 7). CN computes $D_J(C_u)$, N_i'', and checks $H(h_1 \parallel ID_{SN_i} \parallel n_1] = D_{N_i''}[D_i]$, h_1 Steps(8-10). Similarly, CN stores tuple $< ID'_{SN_i}, PID'_u >$ in database with respect to accessed to SN (Step 11). Next, CN computes A_u, B_u. Finally, sends $< M_2$ to sensor node (Steps 12-14). once $< M_2$ received by the SN, it examines timestamps (Step 15). Further, SN computes parameters and verifies ID'_{SN_i} and t_2' (Step 16 and 17). Next, SN computes parameters and verifies h_2 (Step 18 and 19). Additionally, SN chooses a random number and computes parameters $F_s, SK_{U,SN}, G-s$ (Steps 20-23). Finally, SN forwards the message to CN (Step 24). The CN validates timestamp and validates message (Step 25). Similarly, CN computes parameters and checks ID''_{SN_i} and PID''_u (Step 26 and 27). Then CN computes n_2', C_u^n, and L_{CN}. Finally, sends message $< M_4 = L_{CN}$, $t_4 >$ to U through public channel (Steps 29-32). consequently, User verifies the timestamp and validates message (Step 33). In sequel, User calculates parameter and verifies ID'_{SN_i} and t_4 (Step 34 and 35). Futhermore, User computes parameters n_2'', $SK_{U,SN}$ and checks $H(SK_{U,SN})$, h_5 (Step 36 and 37). Then, U authenticates SN_i and stores session key $SK_{U,SN}$ for further secure communication. Similarly, SN_i stores a session key $SK_{U,SN}$ which is shared with User U for further secure communication (Step 38 and 39). The login and mutual authentication process depicted in Table 4.

E. DYNAMIC SN ADDITION PHASE

When sensor nodes captured physically by an adversary or malfunctioning owing to power consumption difficulties since they are battery-powered, new, fresh sensor nodes must be installed in a WSN. As a result, this phase allows for the integration of a few new sensor nodes into the present WSN, allowing users to take use of the services provided by these nodes. Assume that a new sensor node, SN_i^{new}, will be added to the current WSN shortly. Initially, SN_i^{new} forwards registration request with ID_{sn}^{new} to RA. Consequently, RA computes pre-shared secret $K_{sn}^{new} = H(ID_{ra} \parallel ID_{sn}^{new} \parallel MSK_Q)$ and stores ID_{sn}^{new}, K_{sn}^{new} in T_{SN}. Parallely, RA preloads SN_i^{new} with ID_{sn}^{new}, K_{sn}^{new} prior to deployment of them in WSN environment. Later, RA shares information about all newly joined sensor nodes to users.

F. USER CREDENTIAL UPDATE PHASE

When the particular user wishes to update his or her password by entering the ID_u and PW_u without involving RA. initially, user computes $PID_u = H(ID_u \parallel R_u)$, $PWD_u = H(PW_u \parallel R_u)$, $W_S = X_S \oplus PWD_u$, and verifies parameter $C_u = H(ID_u \parallel W_S)$ holds or not, then login process is permitted. Then, user retrieves $R_S = W_S \oplus PWD_U$ with password PW_u. Additionally, chooses another password, which is new one, PW_u^* and calculates $PWD_u^* = H(PW_u^* \parallel R_u)$, $X_S^* = W_S \oplus PWD_u^*$, $R_S^* = Q_S \oplus PWD_u^*$. The user promptly updates parameters in "tamper-proof onboard memory."

V. SECURITY ANALYSIS

A. INFORMAL SECURITY ANALYSIS

The suggested protocol's security analysis may be provided, emphasizing the latter's security and privacy resilience in the face of notable security threats.

Proposition 1: The proposed protocol mitigates User impersonation attacks.

Proof: $\mathcal{A}$ should attempt a login process message $M_1 = C_u, D_i, t_1$ to masquerade a valid user. He or she must, however, complete the login process before initiating an authentication request. During the login process, User should compute $PID_u = H(ID_u \parallel R_u)$, $PWD_u = H(PW_u \parallel R_u)$, $W_S = X_S \oplus PWD_u$, and verifies $V_i = H(PID_u \parallel R_u)$. $\mathcal{A}$ Unless the correct credentials are provided, the adversary would be unable to proceed to the next level. The chances of getting the correct ID_u and PW_u numbers, on the other hand, are vanishingly small.

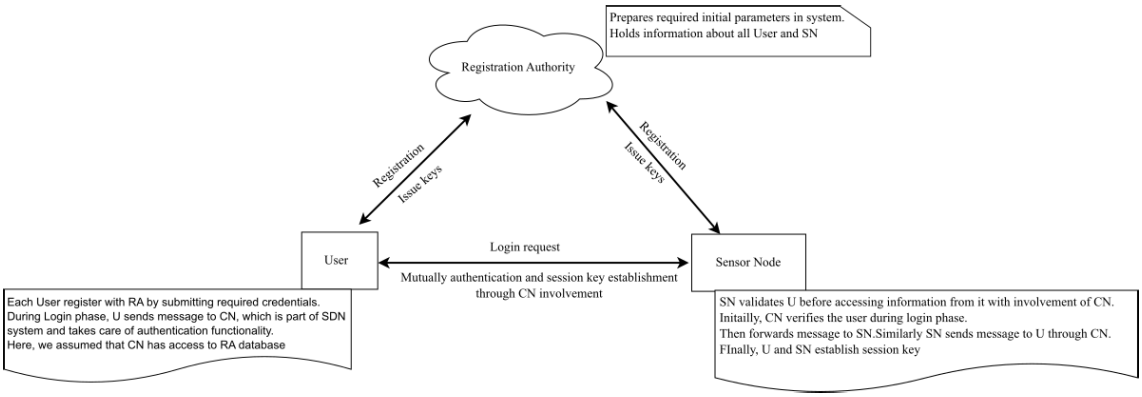

FIGURE 2. Message flow of proposed protocol.

Proposition 2: The proposed protocol is capable of surviving CN impersonation attacks..

Proof: Consider the following scenario: A registered CN_i masquerading as $\mathcal{A}$ obtains $< M_1 >$ and tries to impersonate a valid CN_i with forwarding message $< M_2^{AE} >$ in response. When ID_{CN_i} considered to be with $\mathcal{A}$, *s/he* will have a difficult time obtaining $PID_u = ID_u \parallel R_u$ and n_1 with message M_1. The B_S parameter of targeted *CNi* is mandatory while computing the response M_2^{AE}, since each CN_i holds an unique long-term key K_{SN_i}, calculated using ID_{ra} and ID_{SN_i} as $K_{SN_i} = H(ID_{ra} \parallel ID_{SN_i} \parallel MSK_{SN_i})$. As a result, *AE* is unable to calculate K_{SN_i} and respond to User.

Proposition 3: The proposed protocol is resilient against man in the middle attacks.

Proof: Whenever data is transferred across a shared communication link and $\mathcal{A}$ understands the messages exchanged over the link, then, ''a Man-In-the-Middle (MIM) attack'' on the system is possible. The User, CN, and SN transmits messages on the public channel M_1, M_2, M_3, M_4 during the authentication procedure. As a result, *mathcalA* can collect these data in order to comprehend messages sent between User-CN and CN-SN. The parameters in the above messages are created with ''one-way hash function'', according to the suggested protocol. As a result, $\mathcal{A}$ is unable to locate any critical information that could be, however, used to interpret the communication among the User-CN and CN-SN, and these messages should be computed session by session. As a result, the proposed protocol is capable to counter the ''MIM attacks.''

Proposition 4: The proposed protocol can protect against replay attacks.

Proof: mathcalA could try to start a new session by forwarding messages such as $< M_1, M_2, M_3, M_4 >$. Each communication, however, was timestamped. Before moving on to the next stage, each entity verifies the timestamp and validates the message. Furthermore, to prevent replay attacks, the suggested protocol used random integers such as R_u, n_1, n_2. In this approach, the proposed protocol should prevent replay attacks.

Proposition 5: The proposed protocol is resilient against privileged insider attacks.

Proof: Neither the raw credentials nor the digest of credentials are submitted to *CNi* during the user registration process. The user provides $PID_u = H(ID_u \parallel R_u)$, $PWD_u = H(PW_u \parallel R_u)$ to RA, where $R_u \in Z_P^*$. As a consequence, an insider could not access credentials of any legitimate User. Furthermore, the proposed protocol is predicated on the concept of not maintaining password verification tables, with entity authentication achieved by message validation such as $V_i = H(PID_u \parallel PWD_u \parallel R_u)$. Hence, the proposed protocol is resilient against insider attacks.

Proposition 6: The proposed protocol can withstand password guessing attacks.

Proof: The use of random secret R_u protects both the identification ID_u and the password PW_ui. As a result, without knowing R_u, it is very difficult to obtain user's identity and password. Hence, password guessing attacks are not possible with the proposed protocol.

Proposition 7: The proposed protocol supports user anonymity and untraceability.

Proof: Users are truly hesitant to share any private information, including user identification and needed criteria, in order to reveal their actions and concealed messages. If $\mathcal{A}$ should obtain private information about a registered user, the communication network will be unable to guarantee user anonymity when it comes to registered users. During the login and authentication phases, U, CN, and SN exchange messages $< M_1, M_2, M_3, M_4 >$. All parameters vary for each session due to adoption of n_1, n_2, R_u, and n_3. Similarly, the genuine identity of the user, ID_u, is encoded as $PID_u = H(ID_u \parallel R_u)$. A secure hash method is used to produce all of these values. These generated values provide no information due to their irreversible nature. Hence, The proposed protocol supports user anonymity throughout all phases, as discussed in above. Additionally, The proposed protocol includes an important feature known as untraceability. Since these configurations are random, it's indeed undisclosed and unidentifiable to the adversary.

TABLE 4. Login and mutual authentication phase.

User/SC
1) Enters identity and password ID_u, PW_u
2) computes $R_u = Y_u \oplus H(ID_u \parallel PW_u), PID_u = H(ID_u \parallel R_u), PWD_u = H(PW_u \parallel R_u), W_S = X_S \oplus PWD_u$
3) Checks that $V_i = H(PID_u \parallel PWD_u \parallel R_u)$
4) Computes $N_i^! = N_i^* \oplus H(R_u \parallel PWD_u)$
5) Selects a random number $n_1 \in Z_p^*$, computes $D_i = E_{N_i^!}[H(PID_u \parallel C_u \parallel ID_{SN_i} \parallel n_1 \parallel t_1) \parallel ID_{SN_i} \parallel n_1], Q_S = R_S \oplus PID_u$
6) Sends the login request message $M_1 = C_u, D_i, t_1$
CN
7) Checks timestamp $t_1^* - t_1 \leq \Lambda T$, then proceeds further, otherwise abort
8) Computes $(r_a^{'} \parallel PID_u^{'} \parallel ID_{ra}^{'}) = D_J(C_u), N_i^{''} = H(PID_u \parallel ID_{ra} \parallel K) \oplus PWD_u$
9) $H(h_1 \parallel ID_{SN_i} \parallel n_1] = D_{N_i^{''}}[D_i]$
10) Verifies $h_1 = H(PID_u^{'} \parallel C_u^{'} \parallel ID_{SN_i}^{'} \parallel n_1^{'} \parallel t_1)$, then proceed further, otherwise abort
11) Stores tuple $< ID_{SN_i}^{'}, PID_u^{'} >$ in database with respect to accessing SN
12) Computes $A_u = n_1^{'} \oplus H(K_{SN_i} \parallel PID_u \parallel ID_{SN_i}^{'} \parallel t_2)$
13) $B_u = E_{K_{SN_i}}[h(PID_u \parallel ID_{SN_i}^{'} \parallel n_1^{'} \parallel t_2) \parallel PID_u \parallel ID_{SN_i}^{'} \parallel A_u \parallel t_2]$
14) Sends $< M_2 = B_u, t_2 >$ to sensor node SN_i
SN
15) Checks timestamp $t_2^* - t_2 \leq \Lambda T$, then proceeds further, otherwise abort
16) Calculates $(h_2 \parallel PID_u^{''} \parallel ID_{SN_i}^{''} \parallel A_u^{'} \parallel t_2^{'}] = D_{K_{SN_i}}[B_u]$
17) Verifies $ID_{SN_i}^{'}$ and $t_2^{'}$
18) Calculates $n_1^{'} = A_u^{'} \oplus H(K_{SN_i} \parallel PID_u^{''} \parallel ID_{SN_i} \parallel t_2)$
19) Verifies $h_2 = (PID_u^{''} \parallel ID_{SN_i} \parallel n_1^{''} \parallel t_2)$, Then SN_i confirms the authenticity of CN_i
20) Selects a random number n_2
21) Calculates $F_s = n_2 \oplus H(K_{SN_i} \parallel PID_u^{''} \parallel ID_{SN_i} \parallel t_3)$
22) $SK_{U,SN} = H(PID_u^{''} \parallel ID_{SN_i} \parallel n_1^{''} \parallel n_2)$
23) $G_s = E_{K_{SN_i}}[H(PID_u^{''} \parallel ID_{SN_i} \parallel n_2 \parallel t_3) \parallel H(SK_{U,SN} \parallel PID_u^{''} \parallel ID_{SN_i} \parallel F_s \parallel t_3]$
24) Sends message $< M_3 = G_s, t_3 >$ to controller node CN_i
CN
25) Verifies its validity by checking the timestamp, then continues; otherwise, it aborts
26) Computes $[H(PID_u^{''} \parallel ID_{SN_i} \parallel n_2 \parallel t_3) \parallel H(SK_{U,SN} \parallel PID_u^{''} \parallel ID_{SN_i}^{''} \parallel F_s^{'} \parallel t_3^{'}] = D_{K_{SN_i}}(G_s)$
27) Checks $ID_{SN_i}^{''}$ and $PID_u^{''}$
28) Computes $n_2^{'} = F_s^{'} \oplus H(K_{SN_i} \parallel PID_u^{''} \parallel ID_{SN_i} \parallel t_3)$
29) Chooses a random number n_3
30) Computes $C_u^n = E_J[n_3 \parallel PID_u \parallel ID_{ra}], n_t = n_1^{'} \oplus n_2^{'}$
31) $L_{CN} = E_{N_i}^{''}[H(PID_u \parallel C_u^n \parallel ID_{SN_i}^{'} \parallel H(SK_{U,SN}) \parallel n_2^{'} \parallel t_4) \parallel PID_u \parallel C_u^n \parallel ID_{SN_i}^{'} \parallel n_t \parallel H(SK_{U,SN} \parallel t_4]$
32) Sends message $< M_4 = L_{CN}, t_4 >$ to U
User
33) User U_i verifies timestamps and validates the message
34) U_i calculates $D_{N_i}^{''}[L_{CN}] = [h5 \parallel PID_u \parallel C_u^n \parallel ID_{SN_i}^{'} \parallel n_t \parallel H(SK_{U,SN} \parallel t_4]$
35) Checks $ID_{SN_i}^{'}$ and t_4
36) Computes $n_2^{''} = n_1 \oplus n_t^{'}, SK_{U,SN} = H(PID_u^* \parallel ID_{SN_i} \parallel n_1 \parallel n_2^{''})$
37) Finally, U checks $H(SK_{U,SN})$ and $h5 = H(PID_u^* \parallel C_u^{n^*} \parallel ID_{SN_i} \parallel H(SK_{U,SN}^* \parallel n_2^{''} \parallel t_4)$
38) Then, {U} authenticates SN_i and stores session key $SK_{U,SN}$ for further secure communication
39) Similarly, SN_i stores a session key $SK_{U,SN}$ which is shared with User U for further secure communication.

Proposition 8: The proposed protocol ensures forward secrecy.

*Proof:*Key agreement systems ensure that future session keys should not be calculated, even if long-term credentials are disclosed. The session key in the proposed protocol is computed as $SK_{U,SN} = H(PID_u^* \parallel ID_{SN_i} \parallel n_1^{\parallel} n_2'')$. Because *AE* knows n_1 and n_2, extra variables (PID_u, ID_{SN_i}) are required for the session key computation. The pseudo identity of U is PID_u, which may be computed as $PID_u = H(ID_u \parallel R_u)$, where $R_u \in Z_P^*$ is a random integer. In this scenario, PID_u are long-term variables of U, however obtaining/compiling these values is impossible owing to a lack of essential values. The random numbers R_u, n_1 and n_2 differ in each session key. Taking everything into consideration, the suggested protocol meets the forward secrecy criteria.

Proposition 9: Under the CK-adversary model, the proposed protocol is resistant to ESL attacks.

Proof: A session key between U and SN during the login and authentication phase of the proposed protocol is calculated as $SK_{U,SN} = H(PID_u^* \parallel ID_{SN_i} \parallel n_1^{\parallel} n_2'')$. As a result, it is clear that the creation of $SK_{U,SN}$ includes both short-term secret credentials (n_1 and n_2) and long-term secret PID_u. The adversary will not be able to obtain the secret key until both sorts of secret credentials are compromised. As a result, the ESL attack is protected by the CK-adversary model.

Proposition 10: Under certain conditions, the proposed protocol is capable of withstanding DoS attacks.

Proof: The protocol was built with sublime operations (such as Secure Hash Algorithm, bitwise XOR, and concatenation) that require less processing bandwidth. It should degrade receiver performance by flooding numerous messages, high bandwidth utilization, and weariness on the receiver's end. As a result, user may require timely delivery of critical information. When confronted with a DoS attack, the receiver can instantly check its correctness and reject fraudulent messages, ensuring immunity to DoS attacks. However, no assumptions have been made regarding the similarity of active ''DoS attacks'' via physical connection disruptions.

B. FORMAL PROOF USING BAN LOGIC

This section uses BAN logic to demonstrate the formal security of our new proposed protocol [41]. There are guidelines for analyzing message exchange protocols in BAN Logic. It is essential to figure out whether the messages sent over a protocol are reliable, secure, and resistant to eavesdropping. The proposed protocol's mutual authentication was examined using the BAN Logic. The basic BAN Logic notations before describing concepts are depicted in Table 6.

In this proof, we show that both a legal user U_i and an accessed sensor node SN_n in WSN environment mutually authenticate among each other.

1) RULES

There are the following four rules used in the BAN logic:

- Rule(1). Message-meaning rule:

$$\frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \triangleleft \{X\}_K}{P \mid\equiv Q \mid\sim X} \text{ and } \frac{P \mid\equiv P \overset{Y}{\rightleftharpoons} Q, P \triangleleft \langle X\rangle_Y}{P \mid\equiv Q \mid\sim X}$$

- Rule(2). Nonce-verification rule: $\frac{P\mid\equiv\#(X), P\mid\equiv Q\mid\sim X}{P\mid\equiv Q\mid\equiv X}$
- Rule(3). Jurisdiction rule: $\frac{P\mid\equiv Q\mid\Rightarrow X, P\mid\equiv Q\mid\equiv X}{P\mid\equiv X}$
- Rule(4). Freshness-conjuncatenation rule: $\frac{P\mid\equiv\#(X)}{P\mid\equiv\#(X,Y)}$

2) GOALS

The proposed protocol must satisfy the following test objectives in accordance with the analytical steps of the BAN logic to guarantee system security.

- $G_1 : U_i \mid\equiv U_i \overset{SK}{\rightleftharpoons} SN_i$.
- $G_2 : SN_n \mid\equiv U_i \overset{SK}{\rightleftharpoons} SN_i$ Let's assume that SK is shared session key $SK^*_{U_i,SN_i} \left(= SK_{U_i,SN_i}\right)$ between U_i and SN_i for simple understanding of proposed protocol.

3) IDEALIZED FORM

The proposed scheme's idealized forms are arranged as follows. For the sake of simplicity, assume that $SK_{CN,SN_i} = SK_{CS}$.

- From message $M_1 = \{C_u, D_i, t_1\}$, we have

$$U_i \rightarrow CN : \{r_a \| PID_i \| ID_{ra}\}_{MSKj},$$

$$\left\{h\left(PID_i, C_u, I\ D_{SN_i}, U_i \overset{n_1}{\rightleftharpoons} SN_i, t_1\right),\right.$$

$$\left. ID_{SN_i}, U_i \overset{r_i}{\rightleftharpoons} SN_{n_{U_i}} \overset{N'_i}{\longleftrightarrow} CN.\right.$$

- From message $M_2 = \{B_u, t_2\}$, we have

$$CN \rightarrow SN_i : \{h\left(PID_u, I\ D_{SN'_i}, U_i \overset{n_1}{\rightleftharpoons} SN_i, t_2\right),$$

$$PID_u, I\ D'_{SN_i}, A_u, t_2 SN_i \overset{SK_C S^S CN}{\longleftrightarrow}.$$

- From message $M_3 = \{G_i, T\ S_3\}$, we have

$$SN_n \rightarrow CN : \left\{h\left(PID''_u, ID_{SN_i}, U_i \overset{n_2}{\rightleftharpoons} SN_i, t_3\right),\right.$$

$$\left. h\left(SK_{U_i,SN_i}\right), PID''_u, ID_{SN_i}, F_s, t_3\right\}_{SN_i} \overset{SK_{CS}}{\longleftrightarrow} CN.$$

- From message $M_4 = \{L_{CN}, t_4\}$, we have

$$CN \rightarrow U_i : \left\{h\left(PID_u, , C^n_u, I\ D'_{SN_i}, h_4, U_i \overset{n_2}{\rightleftharpoons} Sn_t, t_4\right),\right.$$

$$\left. PID_u, C^n_u, I\ D'_{SN_i}, n_t, h_4, t_4\right\}_{U=U_i \overset{N'_i}{\longleftrightarrow} CN}.$$

4) HYPOTHESES

To analyze the proposed protocol, the following presumptions about the initial state are made:

- $H_1 : U_i \mid\equiv \#(t_4), CN\mid \equiv \#(t_1), CN \mid\equiv \#(t_3), SN_i\mid \equiv \#(t_2)$.
- $H_2 : U_i \mid\equiv U_i \overset{N'_i}{\longleftrightarrow} CN$.
- $H_3 : CN \mid\equiv U_i \overset{N'_i}{\longleftrightarrow} CN$.
- $H_4 : SN_i \mid\equiv SN_i \overset{SK_{C-S}}{\longleftrightarrow} CN$.
- $H_5 : CN \mid\equiv SN_i \overset{SK_{C-S}}{\longleftrightarrow} CN$.
- $H_6 : U_i \mid\equiv (CN_1 \Rightarrow SN_n \mid\sim X)$.
- $H_7 : SN_n \mid\equiv (CN_1 \Rightarrow U_i \mid\sim X)$.
- $H_8 : CN \mid\equiv \left(U_{i^1} \Rightarrow U_i \overset{n_i}{=} SN_i\right)$.
- $H_9 : CN \mid\equiv \left(SN_{n^1} \Rightarrow U_i \overset{n_i}{\rightleftharpoons} SN_i\right)$.
- $H_{10} : U_i \mid\equiv \left(CN_1 \Rightarrow U_i \overset{n_i}{\rightleftharpoons} SN_i\right)$.
- $H_{11} : SN_i \mid\equiv \left(CN_1 \Rightarrow U_i \overset{n_i}{=} SN_i\right)$.

The idealized form of the proposed scheme is analyzed based on the BAN logic rules and assumptions. The primary proofs are stated as follows:

- From message M_1, message meaning rule, freshness rule, nonce verification rule and H_1, we have

$S_1 : CN \mid\equiv U_i\mid \equiv U_i \overset{n_i}{=} SN_i$.

- From S_1, jurisdiction rule and H_8, we obtain,

$S_2 : CN \mid\equiv U_i \overset{n_i}{\rightleftharpoons} SN_i$. - From message M_2, S_2, message meaning rule, freshness rule, nonce verification rule, H_1 and H_7, we get

$S_3 : SN_i\mid \equiv CN\mid \equiv U_i \overset{n_i}{\rightleftharpoons} SN_i$

- From S_3, jurisdiction rule and H_{11}, we have,

$S_4 : SN_i \mid\equiv U_i \overset{n_i}{\rightleftharpoons} SN_i$

- The session key is computed as $SK = SK_{U_i,SN_i} = h\left(PID''_u \left\| ID_{SN_i}\right\| n''_2 \| n_1\right)$. Thus, from S_4, we obtain the goal G_2 as $G_2 : SN_i \mid\equiv U_i \overset{SK}{\rightleftharpoons} SN_i$.

TABLE 5. Security features and functionality comparison with relevant protocols.

Scheme	SFF1	SFF2	SFF3	SFF4	SFF5	SFF6	SFF7	SFF8	SFF9	SFF10	SFF11	SFF12	SFF13
Li et al. [20]	Yes	Yes	Yes	NO	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Li et al. [22]	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes
Banerjee et al. [25]	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Sutrala et al. [27]	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Roy et al. [28]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes
Proposed	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SFF1: User impersonation attacks SFF2: CN/GW impersonation attack SFF3: MIM attack SFF4: Replay attack SFF5:Privilege insider attack SFF6: User anonymity and untraceability SFF7: DoS attack SFF8: Forward secrecy SFF9: Desynchronization attack SFF10: ESL attack SFF11: Mutual authentication SFF12: Dynamic SN addition SFF13: User credential update phase													

TABLE 6. The notations and meaning of BAN logic.

Notations	Meaning
$P \mid\equiv X:$	P believes a statement X or P is entitled to believe X.
$\#(X):$	Formula X is considered as fresh.
$P_1 \Rightarrow X:$	P has jurisdiction over statement X.
$P \triangleleft X:$	P sees the statement X.
$P \mid\sim X:$	P once said the statement X.
$(X, Y):$	X or Y is a part of formula (X, Y).
$\{X\}_K:$	X is encrypted under the key K.
$\langle X\rangle_Y:$	X is combined with the formula Y.
$P \overset{K}{\longleftrightarrow} Q:$	P and Q may use the shared key K to communicate. The key K is good, in that it will never be discovered by any principal except P and Q.
$P \overset{X}{\rightleftharpoons} Q:$	X is secret known only to P and Q, and possibly to principals trusted by them.

TABLE 7. Comparison of communication cost with relevant protocols.

Scheme	No. of messages	Communication cost(in bits)
Li et al. [20]	4	3520
Li et al. [22]	4	3648
Banerjee et al. [25]	3	1248
Sutrala et al. [27]	3	4192
Roy et al. [28]	4	3680
Proposed	4	768

- From message M_3, message meaning rule, freshness rule, nonce verification, jurisdiction rules, H_1 and H_9, we get,

$S_5 : CN \mid\equiv U_i \overset{n_1}{\rightleftharpoons} SN_i$

- From message M_4, S_5, message meaning rule, freshness rule, nonce verification rule, jurisdiction rule, H_1, and H_6, we get, $S_6 : U_i| \equiv CN| \equiv U_i \overset{n_1}{\rightleftharpoons} SN_i$.
- From S_6, jurisdiction rule, and H_{10}, we have, $S_6 : U_i \mid\equiv U_i \overset{n_1}{\rightleftharpoons} SN_n$
- Since the session key SK is computed as $SK = SK^*_{U_i,SN_i} = h\left(PID^*_u \| ID_{SN_i} \| n_1 \| n''_2\right)$, we get the goal G_1 from S_6 as $G_1 : U_i \mid\equiv U_i \overset{SK}{\rightleftharpoons} SN_i$.

Hence, we achieve both goals G_1 and G_2, and as a result, the authentication is proved.

C. FORMAL SECURITY ANALYSIS USING SCYTHER TOOL

Scyther is a "push-button tool" for validating the security characteristics described in security protocols, according to [42]. To put it another way, the protocol might be tested against the Dolev-Yao adversary [34] and the internal adversary [43]. The tool contains compelling features, such as the ability to assess security objectives using claim events, and failed relevant attack graphs that reveal security goals/claims. All protocol assertions are generally rewritten in a script [44].

Scyther accepts four authentication claims and two confidentiality claims. *alive* (aliveness), *Weakagree* (weak agreement), *Niagree* ("non-injective agreement") and *Nisynch* ("non-injective synchronization"), while the secrecy claims are *Secret* and Session Key Reveal (*SKR*) Aliveness means that connection with the target goal is possible. If both the source and destination are aware of the communication, weak agreement is guaranteed. Non-injective agreement requires agreement on the data that is communicated, in addition to the weak agreement. In addition to the non-injective agreement, the messages must be transmitted in a certain order to ensure non-injective synchronization.The authentication claims are organized in a hierarchy (*Nisync*, *Niagree, Weakagree* and *alive*). . *Nisync* ranks 1 in authentication claims. This means that if *Nisync* holds, then all other authentication claims are met [45]. The *SKR* and *Secret* claims, respectively, are used to protect the confidentiality of session keys and data/messages [46]. The demonstration of the claims demonstrates that the adversarial model's security goals are correct. The Scyther tool findings, as shown in Figure 3, prove that it is difficult to attack authentication activities.

VI. PERFORMANCE ANALYSIS

This section compares the proposed protocol to different WSN authentication techniques based on performance parameters such as security features, communication, and computational cost.

A. SECURITY FEATURES COMPARISON

Various security threats are taken into consideration here to evaluate the performance of the suggested protocol here [20], [22], [25], [27], and [28] as security is the main concern. The term "Yes" is an indication of the fact that the protocol supports feature or is immune to an attack, whereas the term "No" indicates otherwise. The proposed protocol is resistant against prominent security attacks, as demonstrated in Table 5. The table 5 also reveals that protocols such as [20] and [22], are prone to replay attacks, whereas all the protocols

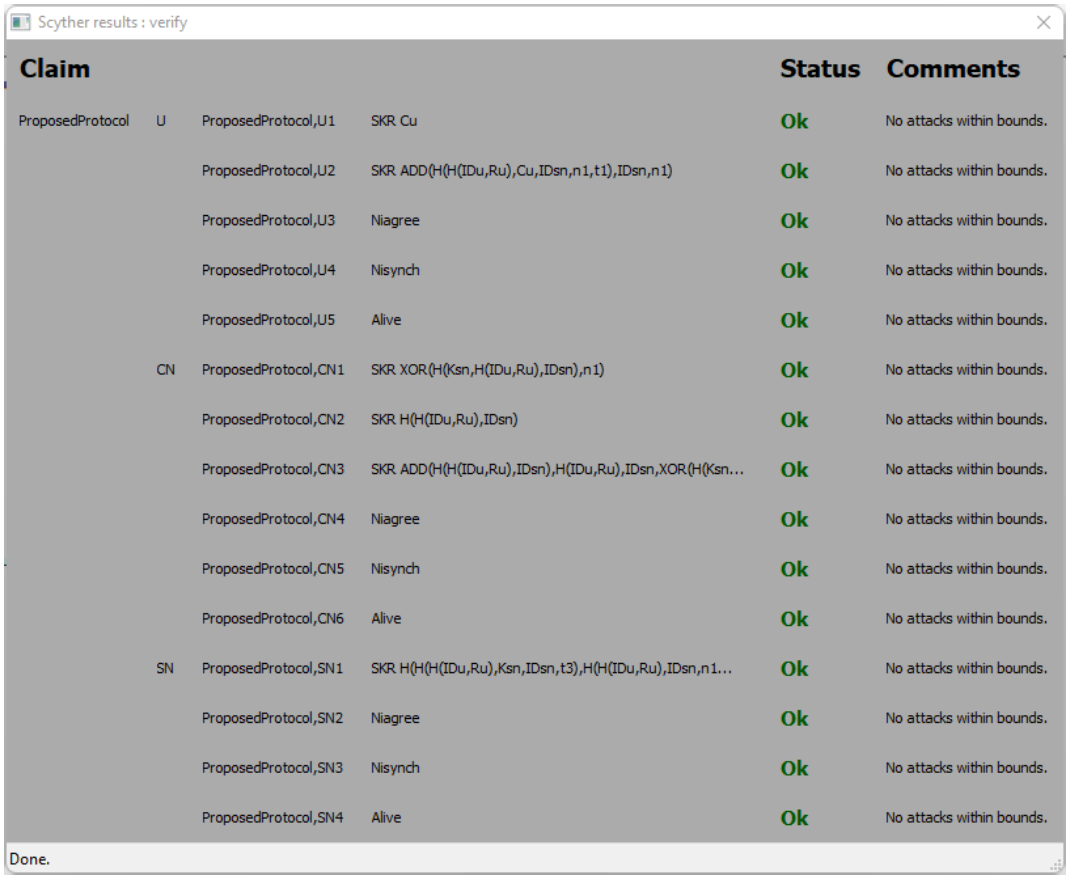

FIGURE 3. Scyther results.

mitigate MIM attacks. The protocol [22] has failed to ensure forward secrecy. Similarly, most of the protocols are immune to desynchronization attacks except [28]. Furthermore, the observation goes that most of the protocols are absconding from the very demanding feature such as dynamic SN addition and user credential updation facility.

B. COMMUNICATION COST COMPARISON

Communication expenses were actually linked to the quantity of data delivered through the channel. It is measured in bits and is defined by the total amount and kind of parameters sent among communication entities in order to permit mutual authentication. This section looks at SHA-256 and AES encryption. We consider a 160-bit identity, a random integer of 160 bits, a 320-bit ECC point, and a timestamp of 32 bits. During the login and mutual authentication phase, U sends message $M_1 =< C_u, D_i, t_1 >$ to CN, which consumes 128 bits, 128 bits, and 32 bits (i.e. 128+128+32 = 288 bits). The CN sends the message to SN as $< M_2 = B_u, t_2 >$, which takes 128 bits and 32 bits (i.e. 128+32 = 160 bits). Then, SN delivers $< M_3 = G_s, t_3 >$ to CN with 128 bits and 32 bits, respectively (i.e 160 bits). Again, CN sends a message $< M_4 = L_{cn}, t_4 >$ to User, which requires 128, and 32 bits, respectively (i.e 160 bits). As a result, the total communication between entities User, CN, and SN throughout the login and authentication phase is 768 bits (i.e. 288+160+160+160). As demonstrated in Table 7, the existing related protocols [20], [22], [25], [27], and [28] need 3520 bits, 3648 bits, 1248 bits, 4192 bits, and 3680 bits, respectively. We can see that the suggested protocol has lower communication costs than all existing related protocols [20], [22], [25], [27], and [28].

C. COMPUTATIONAL COST COMPARISON

The overall quantity and kind of cryptography necessary across communication channels is measured in milliseconds (ms). When the system's computing time is shortened, communications between entities may be delivered more rapidly. We compared the computational expenses at the user device, CN, and SN to other related protocols. The following notations were examined for T_ecm -ECC point multiplication, T_eca -ECC point addition, T_h-one-way hash function, $T_{e/d}$-symmetric encryption/decryption, T_{fe}- feature extraction to determine the computational cost. The computational values for all the notations described above are taken from [27], where the implementation was done on two platforms, server setting and user/smart device configuration. The server environment includes the MacBook Pro model

TABLE 8. Execution times for various cryptographic primitives (in ms) [27].

Notation	Operation	Control Node/Server side	User mobile device/smart device
T_{ecm}	Elliptic curve point multiplication	0.382	2.288
T_{eca}	Elliptic curve point addition	0.002	0.016
$T_{e/d}$	Symmetric encryption/decryption	0.039	0.228
T_{fe}	Feature extraction	0.382	2.288
T_h	Hash function	0.024	0.309

TABLE 9. Computational cost comparison with relevant protocols.

Scheme	User Device	CN/AM/GW	Sensor Node	Total (ms)
Li et al. [20]	$3T_{ecm}+T_{fe}+8T_h \approx 11.624$	$T_{ecm}+7T_h \approx 0.55$	$2T_{ecm}+4T_h \approx 6.136$	$6T_{ecm}+T_{fe}+19T_h \approx 18.31$
Li et al. [22]	$2T_{ecm}+T_{fe}+8T_h \approx 9.336$	$T_{ecm}+9T_h \approx 0.598$	$4T_h \approx 1.236$	$3T_{ecm}+T_{fe}+18T_h \approx 11.17$
Banerjee et al. [25]	$3T_{e/d}+T_{fe}+10T_h \approx 6.062$	$5T_{e/d}+4T_h \approx 0.291$	$2T_{e/d}+2T_h \approx 1.074$	$10T_{e/d}+T_{fe}+16T_h \approx 7.427$
Sutrala et al. [27]	$5T_{ecm}+T_{fe}+2T_{eca}+16T_h \approx 18.395$	$3T_{ecm}+2T_{eca}+9T_h \approx 1.342$	$4T_{ecm}+T_{eca}+8T_h \approx 11.331$	$12T_{ecm}+T_{fe}+5T_{eca}+31T_h \approx 31.068$
Roy et al. [28]	$2T_{ecm}+T_{e/d}+8T_h \approx 7.726$	$12T_h+T_{ecm} \approx 0.67$	$9T_h \approx 2.781$	$3T_{ecm}+T_{e/d}+29T_h \approx 11.177$
Proposed	$2T_{e/d}+8T_h \approx 2.928$	$7T_h+6T_{e/d} \approx 0.402$	$5T_h+2T_{e/d} \approx 2.001$	$20T_h+10T_{e/d} \approx 5.331$

(2019), CPU Architecture: 64-bit, Processor: 2.3 GHz Intel Core i9, Memory: 32 GB, OS: macOS Mojave 10.14.6 as the criterion for evaluating computational costs. Similarly, The user and smart device satisfies the assessment requirements by utilizing a Raspberry Pi 3 B+ Rev 1.3 with a 64-bit CPU, a processor of 1.4 GHz quad-core, 4 cores, and memory (RAM), and an operating system of Ubuntu 20.04 LTS, 64-bit.

From Table 9, we observed that protocol based on public key cryptography such as [27] and [20] incurs more computational cost compared to other relevant protocols. The proposed protocol has a lower communication cost while comparing it with relevant protocols as shown in Table 7. Moreover, the proposed protocol requires less computational cost than other relevant protocols, as shown in Table 9.

VII. CONCLUSION

Our investigation into authenticated key agreement protocols for the WSN environment revealed that most of them had either more performance requirements or were incapable of fulfilling the security requirements. Furthermore, the SDN paradigm has improved upon WSN to avoid a performance bottleneck with traditional network architecture as network traffic, and sensor nodes grow. Hence, we proposed a LAKP for emerging networks leveraging hash functions and XOR operations. Furthermore, the proposed protocol's security needs were assessed formally and informally using the Scyther tool and BAN logic. The proposed protocol protects against well-known attacks such as MIM, replay, impersonation, and insider attacks. The proposed protocol outperforms comparable protocols in terms of computation and communication burdens. The proposed protocol includes innovative features such as dynamic SN insertion and a user credential updating phase. In the future, we plan to propose novel network framework for WSN environment and propose authentication protocol which mitigates physical and machine learning attacks.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.
[2] O. Gnawali, K.-Y. Jang, J. Paek, M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, and E. Kohler, "The Tenet architecture for tiered sensor networks," in *Proc. 4th Int. Conf. Embedded Networked Sensor Syst.*, Oct. 2006, pp. 153–166.
[3] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations," *IEEE Trans. Mobile Comput.*, vol. 11, no. 8, pp. 1399–1411, Aug. 2012.
[4] P. R. Babu, R. Amin, A. G. Reddy, A. K. Das, W. Susilo, and Y. Park, "Robust authentication protocol for dynamic charging system of electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11338–11351, Nov. 2021.
[5] H. Lu, D. Wang, Y. Li, J. Li, X. Li, H. Kim, S. Serikawa, and I. Humar, "CONet: A cognitive ocean network," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 90–96, Jun. 2019.
[6] H. Lu, Y. Zhang, Y. Li, C. Jiang, and H. Abbas, "User-oriented virtual mobile network resource management for vehicle communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3521–3532, Jun. 2021.
[7] Y. Zhang, Y. Li, R. Wang, M. S. Hossain, and H. Lu, "Multi-aspect aware session-based recommendation for intelligent transportation services," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4696–4705, Jul. 2021.
[8] D. Kumar, "A secure and efficient user authentication protocol for wireless sensor network," *Multimedia Tools Appl.*, vol. 80, no. 18, pp. 27131–27154, Jul. 2021.
[9] H. Asgari, S. Haines, and O. Rysavy, "Identification of threats and security risk assessments for recursive internet architecture," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2437–2448, Sep. 2018.
[10] P. R. Babu, B. Palaniswamy, A. G. Reddy, V. Odelu, and H. S. Kim, "A survey on security challenges and protocols of electric vehicle dynamic charging system," *Secur. Privacy*, vol. 5, no. 3, p. e210, May 2022.
[11] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu, "Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the Internet-of-Things," in *Proc. 14th ACM Workshop Hot Topics Netw.*, Nov. 2015, pp. 1–7.
[12] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking,' *Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2014.
[13] P. Sanchez, R. Lopez, and A. Skarmeta, "PANATIKI: A network access control implementation based on PANA for IoT devices," *Sensors*, vol. 13, no. 11, pp. 14888–14917, Nov. 2013.
[14] B. A. A. Nunes, M. A. S. Santos, B. T. de Oliveira, C. B. Margi, K. Obraczka, and T. Turletti, "Software-defined-networking-enabled capacity sharing in user-centric networks," *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 28–36, Sep. 2014.

[15] P. Zhang, C. Lin, Y. Jiang, Y. Fan, and X. Shen, "A lightweight encryption scheme for network-coded mobile ad hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2211–2221, Sep. 2014.
[16] C.-C. Chang and H.-D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 357–366, Jan. 2015.
[17] A. K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, and X. Huang, "Provably secure user authentication and key agreement scheme for wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3670–3687, Nov. 2016.
[18] P. Gope and T. Hwang, "A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7124–7132, Nov. 2016.
[19] S. Steinbrecher and S. Köpsell, "Modelling unlinkability," in *Proc. Int. Workshop Privacy Enhancing Technol.* Cham, Switzerland: Springer, 2003, pp. 32–47.
[20] 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 Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3599–3609, Aug. 2017.
[21] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.
[22] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in Internet of Things environments," *J. Netw. Comput. Appl.*, vol. 103, pp. 194–204, Feb. 2018.
[23] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 4957–4968, Sep. 2019.
[24] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.
[25] S. Banerjee, V. Odelu, A. K. Das, J. Srinivas, N. Kumar, S. Chattopadhyay, and K.-K.-R. Choo, "A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8739–8752, Oct. 2019.
[26] W. Iqbal, H. Abbas, P. Deng, J. Wan, B. Rauf, Y. Abbas, and I. Rashid, "ALAM: Anonymous lightweight authentication mechanism for SDN-enabled smart Homes," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9622–9633, Jun. 2021.
[27] A. K. Sutrala, M. S. Obaidat, S. Saha, A. K. Das, M. Alazab, and Y. Park, "Authenticated key agreement scheme with user anonymity and untraceability for 5G-enabled softwarized industrial cyber-physical systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2316–2330, Mar. 2022.
[28] P. K. Roy and A. Bhattacharya, "SDIWSN: A software-defined networking-based authentication protocol for real-time data transfer in industrial wireless sensor networks," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 3, pp. 3465–3477, Sep. 2022.
[29] X. Li, S. Liu, S. Kumari, and C.-M. Chen, "PSAP-WSN: A provably secure authentication protocol for 5G-based wireless sensor networks," *Comput. Model. Eng. Sci.*, vol. 135, no. 1, pp. 711–732, 2023.
[30] S. Yu and Y. Park, "SLUA-WSN: Secure and lightweight three-factor-based user authentication protocol for wireless sensor networks," *Sensors*, vol. 20, no. 15, p. 4143, Jul. 2020.
[31] S. Frankel, R. Glenn, and S. Kelly, *The AES-CBC Cipher Algorithm and Its Use With IPsec*, document RFC3602, RFC Editor, USA, 2003, doi: 10.17487/RFC3602.
[32] P. R. Babu, A. G. Reddy, B. Palaniswamy, and A. K. Das, "EV-PUF: Lightweight security protocol for dynamic charging system of electric vehicles using physical unclonable functions," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3791–3807, Sep. 2022.
[33] Q. Do, B. Martini, and K.-K. R. Choo, "The role of the adversary model in applied security research," *Comput. Secur.*, vol. 81, pp. 156–181, Mar. 2019.
[34] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.
[35] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 2002, pp. 337–351.
[36] P. R. Babu, A. G. Reddy, B. Palaniswamy, and S. K. Kommuri, "EV-Auth: Lightweight authentication protocol suite for dynamic charging system of electric vehicles with seamless handover," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 734–747, Sep. 2022.
[37] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 8, pp. 769–780, Aug. 2000.
[38] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 2001, pp. 453–474.
[39] B. Palaniswamy, S. Camtepe, E. Foo, L. Simpson, M. A. R. Baee, and J. Pieprzyk, "Continuous authentication for vanet," *Veh. Commun.*, vol. 25, Jan. 2020, Art. no. 100255.
[40] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym schemes in vehicular networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 228–255, 2nd Quart., 2015.
[41] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.
[42] C. J. Cremers, "The Scyther tool: Verification, falsification, and analysis of security protocols," in *Proc. Int. Conf. Comput. Aided Verification*. Cham, Switzerland: Springer, 2008, pp. 414–418.
[43] D. Basin and C. Cremers, "Know your enemy: Compromising adversaries in protocol analysis," *ACM Trans. Inf. Syst. Secur.*, vol. 17, no. 2, pp. 1–31, Nov. 2014.
[44] C. J. F. Cremers, *Scyther: Semantics and Verification of Security Protocols*. Eindhoven, The Netherlands: Eindhoven Univ. Technol., 2006.
[45] C. J. F. Cremers, S. Mauw, and E. P. de Vink, "Injective synchronisation: An extension of the authentication hierarchy," *Theor. Comput. Sci.*, vol. 367, nos. 1–2, pp. 139–161, Nov. 2006.
[46] C. J. Cremers and S. Mauw, "Checking secrecy by means of partial order reduction," in *Proc. Int. Workshop Syst. Anal. Modeling*. Cham, Switzerland: Springer, 2004, pp. 171–188.

NEMALIKANTI ANAND (Graduate Student Member, IEEE) received the Bachelor of Technology degree in computer science and engineering from Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India, and the Master of Technology degree in network and internet engineering from Pondicherry Central University, Puducherry, India. He is currently pursuing the Doctoral (Ph.D.) degree with the School of Computer and Information Sciences, University of Hyderabad, Hyderabad, Telangana, India. His research interests include computer networks and cyber security.

M. A. SAIFULLA (Member, IEEE) received the M.S. degree in computer science and engineering from the Indian Institute of Technology, Madras (IIT Madras), in May 2003, with a focus on computer networks, and the Ph.D. degree in computer science and engineering from Anna University, Chennai (AUC). At IIT Madras, he was a Project Officer with the TeNeT Group (which developed award-winning products). He worked in research and development divisions of top software organizations for around a decade and was a part of innovative product development. His last software research and development job was with Cisco Systems, Bangalore, where he was a part of the team which developed the content or server load balancer. Currently, he is an Assistant Professor with the School of Computer and Information Sciences (SCIS), University of Hyderabad. He presented multiple papers in reputed national and international national conferences and published multiple papers in peer-reviewed international journals. His research interests include network traffic analysis, data center products, network management, software-defined networking, named data networking, and quantum communications. He received a couple of patent submission awards, in 2006 and 2007, for his innovative work. He also received the Cisco Star Award, in 2011, for his outstanding performance in software development. He received the Gold Medal for academic excellence from the M.Edu. Society, Hyderabad.