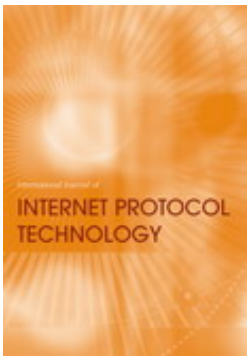


International Journal of Internet Protocol Technology

ISSN online: 1743-8217 - ISSN print: 1743-8209
https://www.inderscience.com/ijipt

Practical and scalable access control mechanism for wireless sensor networks

Ummer Iqbal Khan, Ajaz Hussain Mir

DOI: 10.1504/IJIPT.2023.10054904

Article History:

Received:	27 January 2020
Last revised:	15 September 2020
Accepted:	03 June 2021
Published online:	23 March 2023

Practical and scalable access control mechanism for wireless sensor networks

Ummer Iqbal Khan* and
Ajaz Hussain Mir

Department of Electronics and Communication Engineering,
National Institute of Technology (Srinagar),
Srinagar, Jammu & Kashmir, India
Email: ummer.iqbal.khan@gmail.com
Email: ahmir@rediffmail.com
*Corresponding author

Abstract: The access control mechanism is a necessary security primitive for deploying a new node within the resource-constrained WSN. In literature, new node access control schemes have been proposed for WSN, focusing on efficiency and security strength. However, less attention is given to the functional specification of scalability and independence from time synchronisation. In this paper, an ECC-based new node access control is presented. Besides being computationally effective and secure, the scheme is scalable and doesn't have time synchronisation issues. The proposed scheme's security strength and correctness have been proven using BAN logic and the Random Oracle Model. Simulations on AVISPA and Scyther tools have been performed for automatic security verification of the proposed method. The proposed scheme has also been programmed on TinyOS to perform simulation on the TOSSIM simulator and test-bed implementation on MicaZ motes.

Keywords: access control; AVISPA; BAN logic; elliptical curve cryptography; random Oracle model; Scyther; TinyOS; TinyECC; wireless sensor network; authentication.

Reference to this paper should be made as follows: Khan, U.I. and Mir, A.H. (2023) 'Practical and scalable access control mechanism for wireless sensor networks', *Int. J. Internet Protocol Technology*, Vol. 16, No. 1, pp.11–33.

Biographical notes: Ummer Iqbal Khan received his Bachelor of Engineering degree in Computer Science & Engineering from Dayananda Sagar College, Vishweriaya Technical University Bangalore in 2007. From 2007 to 2010, he served in HCL Technologies as Software Engineer. He joined the National Institute of Electronics and Information Technology in 2010 and is currently working as Scientist 'C' at NIELIT Srinagar/Jammu. He completed his MTech degree in Communication and Information Technology (CIT) from the National Institute of Technology Srinagar, India and is currently pursuing his PhD degree in Wireless Sensor Networks and IoT at the National Institute of Technology (NIT), Srinagar, India. His research interests include wireless sensor networks, IoT, network security and information security & open source technologies.

Ajaz Hussain Mir is a Professor in the department of Electronics and Communication Engineering at National Institute of Technology, Srinagar. He received his BE degree in Electrical Engineering with specialisation in Electronics and Communication Engineering, MTech and PhD degrees in Computer Technology from the IIT Delhi, India in 1989 and 1996, respectively. He is a Chief Investigator of Ministry of Communication and Information Technology, Govt. of India project: Information Security Education and Awareness (ISEA). He has been guiding PhD and MTech theses in digital image processing, computer networks and other related areas and has a number of international publications to his credit. His research interests include biometrics, image processing, security, wireless communication and networks.

1 Introduction

Wireless Sensor Network (WSN) is an active area of research as it serves as a primary data source in smart city applications (Bukhari et al., 2018; Abdmeziem et al., 2015). By 2020, the market for Wireless Sensor Networks is expected to develop at an exponential rate of more than two billion dollars (Kim and Hong, 2013). Typical applications of WSN include environmental monitoring, automation in industry, surveillance, etc. The limitation of resources makes a predominant contrast between WSN and traditional networks. As a result, most of the solutions or protocols of conventional networks cannot be directly applied to WSN (Perrig et al., 2004).

The security of the sensor network is of paramount importance. However, traditional security protocols are not directly applicable to these networks as they have significant resource limitation. Besides having severe communication and computational limitation, unreliable communication also serves as a bottleneck in designing security protocols for WSN (Malan et al., 2008). Moreover, the unattended operation of WSN makes the nodes in the network vulnerable to physical capture attacks (Mo and Chen, 2019). Any security protocol targeting WSN must oblige to its constraints (Gura et al., 2004). Confidentiality, integrity, availability, data freshness and authentication are the core security needs of WSN. The access control mechanism is a bedrock for various security requirements in WSN.

The deployment of a new WSN node is regulated by an access control mechanism. The need for the re-deployment of the nodes arises as their battery may drain out or might get attacked by adversaries. A malicious node deployment may result in complete disruption of the network operation (Parno et al., 2005). A new node access control scheme is responsible for the following two functions (Chatterjee and Das, 2014):

1 *Dynamic node authentication*: It entails an authentication mechanism between the newly deployed node and its adjacent nodes to verify the authenticity of the newly deployed node so as to become the part of the network.

2 *Key exchange*: It entails an establishment of a pair-wise symmetric key of a new node with its neighbours. The established shared key serves as a foundation for other major security primitives.

Authentication and symmetric key establishment schemes have been proposed (Chan and Perrig, 2003; Eschenauer and Gligor, 2002; Karlof et al., 2004). However, they are not dynamic in nature. Aside from being resource-efficient, a new node access control method must also meet some basic security and functional specifications for practical consideration.

1.1 Major security specifications for new node access control scheme

a) *Defence against false data injection attack:* An attacker can eavesdrop on the communication between the sender and receiver nodes and instil false data, resulting in sensor data masquerading. Lack of authentication and confidentiality between the communicating nodes from the basis of this attack. To thwart this attack between any two communicating WSN nodes, data sent between them must be kept secure and authorised. Following the addition of a newly deployed node to WSN, a new node access control mechanism must also set up a shared symmetric key between the deployed node and its adjacent neighbours for ensuring the confidentiality and authentication of messages (Jamalipour and Zheng, 2007).

b) *Defence against node compromise attacks:* A node compromise attack comprises of capturing a group of nodes in a network and obtaining key security information from them. The information extracted may jeopardise the security of the entire network. Assume that N_C is a collection of nodes contained within a network having N_T nodes. If data extracted from N_C nodes does not affect the security of the network's $N_T - N_C$ nodes, the design of the scheme is generally robust to node compromise attack.

c) *Defence against Sybil attack:* In a Sybil attack, the deployed Sybil nodes are capable of claiming several identities, causing network operations to fail. A new node access control scheme can withstand this attack by preventing malicious node deployment.

d) *Defence against wormhole attack*: In a wormhole attack, two malicious sensor nodes tunnel data packets between themselves in order to create a diversion in the WSN. The deployed rogue node will ignore any adjacent neighbours, causing network routing to be disrupted. An access control scheme must prevent such nodes from being deployed in the network (Hu et al., 2006).

e) *Defence against man in the middle (MITM) attack*: A Man in the Middle (MITM) attack occurs when an attacker positions himself into a communication between two sensor nodes. In a MITM attack between nodes (ND_I, ND_J) an attacker creates a malicious pair-wise key with ND_I and ND_J so that the attacker can intercept, modify and masquer traffic between them (Jamalipour and Zheng, 2007.).

f) *Defence against replay attack*: An attacker can gain unauthorised network access in a replay attack by replaying the old access control messages. The design of the new node access control must be impenetrable to such attacks.

1.2 Major functional specifications for new node access control

1) *Low overheads:* To function effectively in resource-constrained WSN, an access control mechanism must be computationally and communicationally efficient. Typically, an access control mechanism should require fewer bits to be broadcast and receive, as communication consumes three times the energy as required for processing (Carman et al., 2000).

2) *Scalable to support large WSN:* The scalability of an access control mechanism requires that the base station is not involved in new node addition. The Base Station's involvement in new node deployment enhances the computational and communication overhead as the size of the WSN increases (Chatterjee and Roy, 2018), limiting the network's scalability.

3) *Independent on time synchronisation issues*: To guard against the replay attack, the new node access control scheme should ideally be independent of time synchronisation between network nodes. Many existing access control mechanisms in the literature rely on timestamps to ensure the freshness of messages exchanged. Owing the high resource overhead associated with traditional time synchronisation schemes such as Network Time Protocol, the access control mechanism's reliance on time synchronisation increases its overheads and complexity (Lasassmeh and Conrad, 2010).

1.3 Contribution

The major contributions are given as below:

1 The proposed scheme presents the design of an ECC-based new node access control scheme for WSN with support for all essential functions and security requirements at a better trade-off than existing related schemes.

2 The proposed scheme supports a large WSN.

3 The designed scheme does not necessitate time synchronisation within the nodes in a network.

4 The security strength and correctness of the scheme have been proven using BAN logic (Burrows et al., 1989) and the Random Oracle Model.

5 A simulation study on AVISPA (Armando et al., 2005) and Scyther (Cremers, 2008) tools have been carried out to verify and validate the security strength of the proposed protocol.

6 Practical implementation of the scheme has been carried out on TinyOS (Levis and Gay, 2009) using TinyECC (Liu and Ning, 2008). The scheme has been simulated using TOSSIM (Levis and Gay, 2009) to estimate energy consumed while considering a large WSN. Moreover, a MicaZ (Mote works, 2013) based small tested implementation has been performed to analyse the scheme's working on practical WSN motes.

1.4 Paper organisation

The remainder of the paper is laid out as follows. The highlights and limitations of the existing schemes have been presented in Section 2. The new node access control scheme is presented in Section 3. Sections 4 and Section 5 gives a thorough informal analysis of the functional and security features of the proposed scheme. In Section 6, security analysis using BAN logic has been carried out. Section 7 provides the details of formal security analysis using the Random Oracle Model. Section 8 provides the simulation details of formal security analysis using AVISPA and Scyther tool. Section 9 presents the comparative summary of the proposed scheme with other relevant schemes in terms of functional and security specifications. Section 10 offers the practical implementation details using TinyOS and TOSSIM Simulator. Finally, Section 11 provide conclusion.

2 Literature review

(Zhou et al., 2007) proposed an access control protocol based on Elliptical Curve Cryptography (ECC). The scheme is based on a preloaded certificate used by a new node to prove its identity to its neighbours. The scheme also provides a mechanism for establishing a symmetric key between two neighbouring nodes in a network. The scheme had a high computational cost as it involves three scalar multiplications and 20 message transmissions for achieving authentication and shared key establishment. However, the scheme highlighted the use of ECC for access control in WSN. The scheme focused on preventing malicious nodes from joining the network rather than detecting them once they have become part of the network. The scheme is scalable. However, require the use of clock synchronisation.

The Novel Access Control Protocol (NACP) was proposed by Haung (2009). The scheme is designed using a hash chain and ECC. The scheme is computationally efficient; however, it cannot be considered for practical purposes. This because the scheme requires the intervention of WSN gateway during new node addition, thus limiting its scalability. Haung (2009) also had other issues which include lack of hash chain renewability, replay attack and new node masquerading attack. NACP does not require clock synchronisation. (Hyun-Sung and Sung-Woon, 2009) proposed Enhanced Novel Access Control Protocol (ENACP) to overcome the limitation of NACP. The scheme addressed the hash chain renewability issue in NACP. The scheme also addressed the security issues of NACP, which include replay attacks and new node masquerading attacks. ENACP is also not scalable as it is dependent on Base Station for new node deployment. Zeng et al. (2010) and Shen et al. (2010) highlighted that ENACP has a significant security and functional limitations.

Haung (2011) suggested a new scheme based on ECC and hash functions. Huang (2011) compared Zhou et al. (2007) and evidently decreased the no of transmission by half, and computational overhead was mainly reduced by one scalar multiplication. In Huang (2011), if an adversary captures a node and extracts the information from it, it can deploy a rogue node in a network by using the same information, making it vulnerable to node capture attacks. The scheme also had a high computational overhead for practical consideration. Lee et al. (2012) proposed the Practical Access Control Protocol (PACP), emphasising that an access control protocol must be stateless in order to be practical. A WSN node operates in two modes: active and sleep. A WSN node that is in sleep mode is unable to receive packets. As such, no hash chain-based scheme,

including NACP and ENACP, can be considered for practical implementation because they require network state maintenance. Practical access Control Protocol (PACP) comprises of two schemes SecPACP and ePACP. PACP has a high memory overhead as it depends on the size of the network. This leads to high overhead and inhibits its scalability.

Das et al. (2013) suggested a scheme for access control for large distributed WSN networks. In this scheme, a preloaded certificate was embedded in a node along with certificate serial, bootstrapping time and version. However, the scheme had a high computational overhead. Though the scheme is scalable, its practical consideration is limited due to the need for clock synchronisation. Das et al. (2013) pointed out that Huang's (2011) scheme is vulnerable to an active attack known as Man in the Middle Attack. (Chatterjee and Das, 2014) suggested a scheme to overcome the limitation of the Huang (2011) scheme. However, the scheme suffered from high computational overhead. Chatterjee et al. (2015) suggested a scheme based on bitwise XOR and hash. The scheme has a significantly very low computational overhead. However, the scheme has scalability issues. Chatterjee and Roy (2018) proposed a scheme with high scalability. However, the scheme has practical concerns in terms of time synchronisation. The scheme is secure against various attacks and has been formally validated using automated tools.

3 Proposed scheme

This section has presented a practical and scalable new node access control scheme based on ECC and one-way collision resistant hash. Table 1 lists the notations used in the scheme.

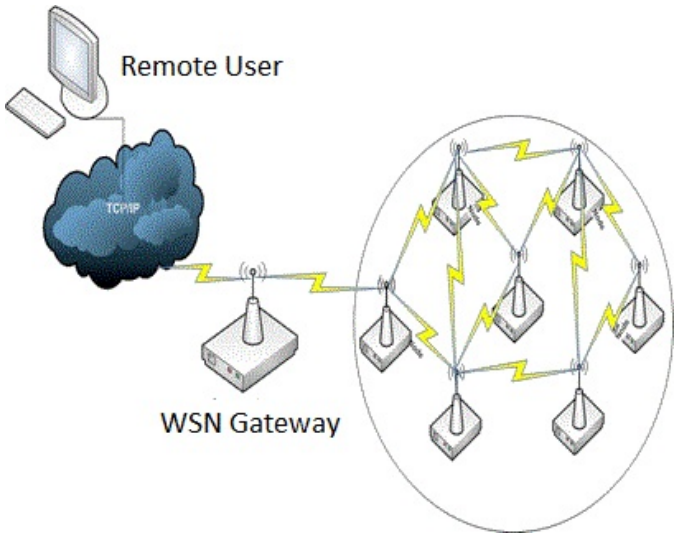
3.1 Network and threat model

A distributed network topology is considered, as shown in Figure 1. The sensor nodes are randomly set out in the area to be monitored and communicate to BS over a multi-hop network. Nodes are deployed in the area of interest as a random collection of sets with the unique deployment identity for each node. The deployment identifier helps to prevent replay attacks by distinguishing between old and new deployments of the same node. BS is assumed to be a trusted entity. The Threat Model being considered is Dolev and Yao (1983), which models the communication over an insecure channel. The communicating parties are also not considered to be trusted entities. The packets can be replayed and eavesdropped on. The nodes in the network are susceptible to tampering. The BS is considered to be very resourceful and cannot be compromised.

Table 1 Notations

Notation	*Description*
$E(a,b)$	Elliptical Curve
$H(.)$	Collision Free One Way Hash Function
ND_I	Identifier of Node I
ND_J	Identifier of Node J
K_I	Private Key of ND_I
PU_I	Public Key of ND_I
PU_{base}	Public Key of BS
K_{base}	Private Key of BS
(C_{NI}, S_{NI})	Signature Certificate of ND_I
D_{NI}	Deployment version of ND_I
D_{NI}	Deployment version of ND_I
RN	Nounce
K_{IJ}	Shared Key between ND_I and ND_J
D_{NI}^J	Latest Deployment Version of ND_I in the Access Control List of ND_J
G	Base Point of $E(a,b)$
BS	Base Station
*	Scalar Multiplication

Figure 1 Network topology



3.2 Proposed access control scheme

The designed scheme comprises of the following two phases: (1) Initialisation (2) Node authentication and key establishment.

3.2.1 Initialisation

The initialisation phase is done before the network setup. The base station performs the following steps during initialisation:

1 Base Station selects an Elliptical Curve $E(a,b): y^2 = x^3 + ax + b$ over a finite field F_{pr}. The parameters *a*, *b* are to be chosen such that $x^3 + ax + b$ does not have any repeated factors, alternatively $4ax^3 + 27b^2 \neq 0$.

2 Base Station computes its public key PU_{base}, where $PU_{base} = K_{base}.G$

3 Base Station generates a set of random numbers $S = \{K_1, K_2, ..., K_n\}$ where n is the number of nodes in the network. For each Node ND_I, *BS* generates its private and public key pair (K_I, PU_I) where :

$$PU_I = K_I * PU_{base}$$

4 Base Station computes the signature pair for all the nodes in the network. For a Node ND_I, the signature pair is computed by the additive splitting of K_{base} into two unequal half's (K^1_{base}, K^2_{base}) as:

$$C_{NI} = K_I * K^1_{base} * G$$

$$S_{NI} = K_I * K^2_{base} * H(ND_I \,||\, DV_I) * G$$

For all the nodes deployed as a first set within the network, the deployment version is set to 1.

5 Each node maintains an Access Control List (ACL), tabulating the neighbouring nodes with which the node has undergone the Node authentication and Key establishment phase. The ACL of the node contains the details of the neighbouring Node, which include: node id, pair-wise key and Latest Deployment Version. An entry of a Node ND_I in the ACL of ND_J is given in Figure 2.

Figure 2 ACL entry of ND_I in ND_J

Node id	*Key*	*Latest Deployment Version*
ND_I	K_{IJ}	$D_{NI}{}^J$

6 Each Node ND_I is preinstalled with the following data:

a) $E(a,b)$

b) $H()$

c) C_{NI} and S_{NI}

d) PU_{base}

e) ND_I

f) DN_I

3.2.2 Node authentication and key establishment

In this phase, the newly deployed node is authenticated by its neighbours to become part of the network. A new node also creates a pair-wise key with its neighbours to communicate securely. Assume ND_I is a new node looking to become the part of the network. The new node ND_I starts the access control mechanism so that it can join the network. The new node ND_I broadcasts $(C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I) \,||\, H[C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I)]$ to its neighbouring nodes in the communication range:

$$ND_I \rightarrow *: (C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I) \,||\, H[C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I)]$$

All neighbouring nodes of ND_I receive the message. Assume ND_J is one of the adjacent nodes that receive the ND_I broadcast. During Node Authentication and Key establishment, the following steps are taken between ND_I and ND_J :

1 The neighbouring node ND_J checks whether the New Node ND_I is in its ACL or not. The following possibilities may arise:

Case 1: If ND_I is not in the ACL of ND_J, then the request from ND_I is considered fresh and accepted for further evaluation to determine the authenticity of the request. Subsequently, Step 2 is performed.

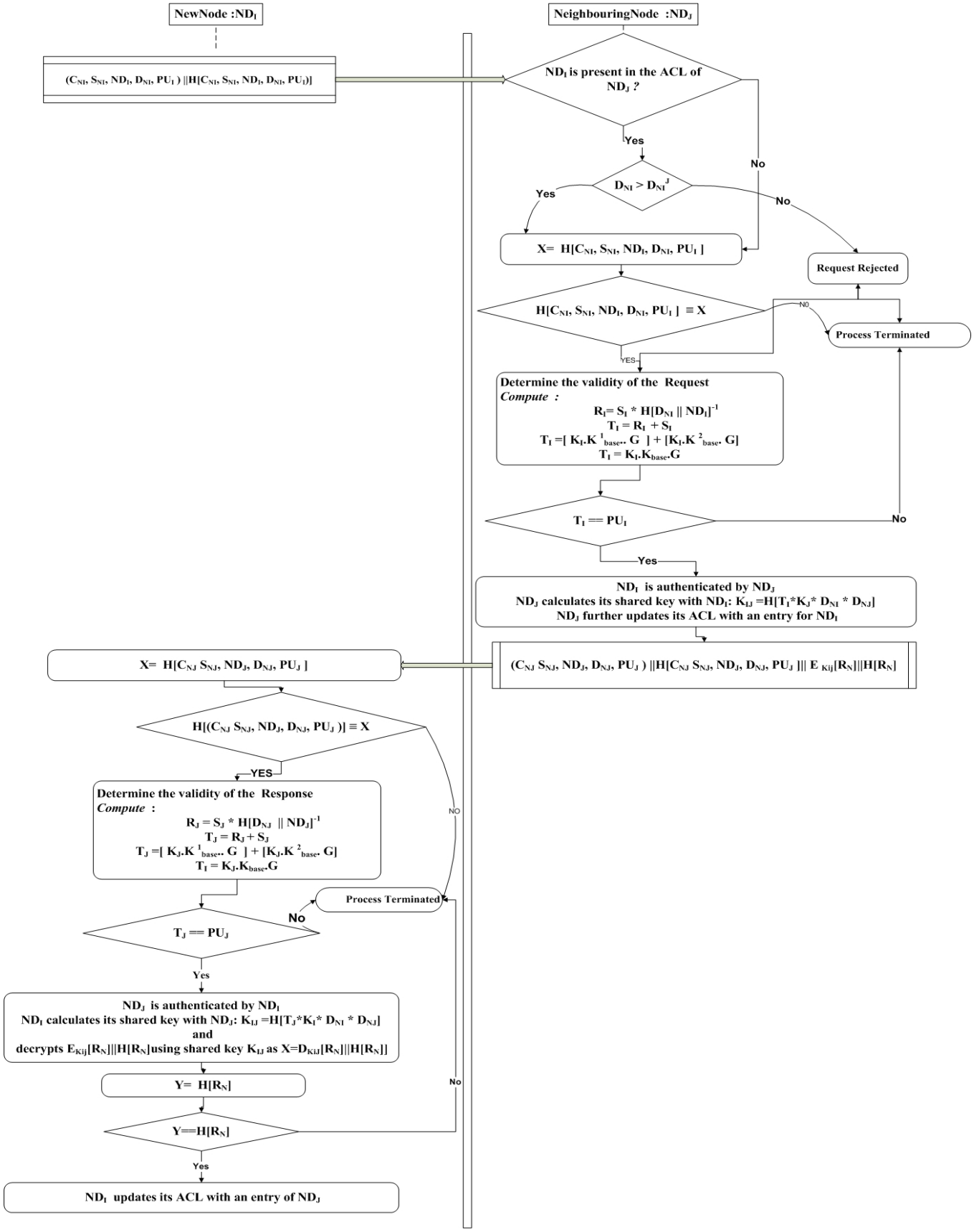
Case 2: if ND_I is present in the ACL of ND_J and $D_{NI} > D^J_{NI}$, then the request from ND_I is considered fresh and accepted for further evaluation and Step 2 is carried out to determine the authenticity of the request. If $D_{NI} <= D^J_{NI}$, then the request is considered obsolete, and the process is aborted.

2 ND_J computes the hash of $(C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I)$ as $X = H[C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I)]$ to check its integrity. The received hash is compared to the hash computed by ND_J :

$$H[C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I] \equiv X$$

If the integrity check returns false, the request is rejected without being processed. Step 3 is done if the integrity check evaluates to be true.

Figure 3 Schematic of new node authentication and key exchange handshake



3 ND_J computes:

$$R_I = S_I * H\left[D_{NI} \mid ND_I\right]^{-1}$$

$$T_I = R_I + S_I$$

$$T_I = \left[K_I.K^1{}_{base} * G\right] + \left[K_I.K^2{}_{base} * G\right]$$

$$T_I = K_I.K_{base} * G$$

If $T_I == PU_I$ is true, ND_J authenticates ND_I and establishes a symmetric key with ND_I.

4 Node ND_J calculates its symmetric key with ND_I as

$$K_{IJ} = H\left[T_I * K_J * D_{NI} * D_{NJ}\right]$$

$$K_{IJ} = H\left[K_I * K_J * K_{base} * D_{NI} * D_{NJ} * G\right]$$

and updates its access control list with an entry for ND_I

5 ND_J sends $\left(C_{NJ}, S_{NJ}, ND_J, D_{NJ}, PU_J\right) \| H\left[C_{NJ}, S_{NJ}, ND_J, D_{NJ}, PU_J\right] \| E_{KIJ}\left[R_N\right] \| H\left[R_N\right]$ to ND_I.
$ND_J \rightarrow ND_I$:
$\left(C_{NJ}, S_{NJ}, ND_J, D_{NJ}, PU_J\right) \| H\left[C_{NJ} S_{NJ}, ND_J, D_{NJ}, PU_J\right] \| E_{Kij}\left[R_N\right] \| H\left[R_N\right]$

6 On receiving $\left(C_{NJ} S_{NJ}, ND_J, D_{NJ}, PU_J\right) \| H\left[C_{NJ} S_{NJ}, ND_J, D_{NJ}, PU_J\right] \| E_{Kij}\left[R_N\right] \| H\left[R_N\right]$, ND_I computes the hash of the received message) as $X = H\left(C_{NJ}, S_{NJ}, ND_J, D_{NJ}, PU_J\right)$ to check its integrity. The received hash is compared to the hash calculated by ND_I :

$$H[C_{NJ} S_{NJ}, ND_J, D_{NJ}, PU_J] \equiv X$$

If the integrity check returns a false result, no processing is performed and the message is rejected. However, if the integrity check is found to be correct, Step 7 is carried out.

7 ND_I computes:

$$R_J = S_J * H\left[D_{NJ} \parallel ND_J\right]^{-1}$$

$$T_J = R_J + S_J$$

$$T_J = \left[K_J.K^1{}_{base} * G\right] + \left[K_J.K^2{}_{base} * G\right]$$

$$T_I = K_J.K_{base} * G$$

If $T_J == PU_J$ is true, then ND_I authenticates ND_J and establishes a symmetric key with ND_J.

8 Node ND_I computes its symmetric key with N_J as:

$$K_{IJ} = H\left[T_J * K_I * D_{NI} * D_{NJ}\right]$$

$$K_{IJ} = H\left[K_I * K_J * K_{base} * D_{NI} * D_{NJ} * G\right]$$

ND_I further decrypts $E_{KIJ}\left[R_N\right] |$ $as\, Z = D_{KIJ}\left[E_{KIJ}\left[R_N\right]\right]$ and compares $H\left[R_N\right]$ with $H\left(Z\right)$.

$$H\left[R_N\right] \equiv H\left(Z\right)$$

if true, ND_I updates its access control list with an entry for ND_J.

The schematic of the process between ND_I and ND_J is shown in Figure 3. The same procedure is carried out between ND_I and its other neighbouring nodes.

4 Functional analysis

a) *Low overheads*: The proposed new node access control scheme is based on ECC and hash functions, making it efficient for resource constraint nodes. To analyse the computational overhead in the designed scheme, the number of computationally intense operations involved in the designed scheme have been considered (Iqbal and Mir, 2020b). The various critical operations considered for evaluation are shown in Table 2. $2T_{EM} + 3T_{HA} + T_{PA} + T_E / T_D + T_{INV}$ are the critical operations involved in the scheme. The number of bits sent and received is calculated to determine the communication overhead. Two messages are exchanged in the proposed method. Table 3 shows the size of the various parameters involved. The total number of bits sent/received is 5248. No bits preloaded in a node during the initialisation are taken into account for calculating memory overhead. The different parameters preloaded in the node during initialisation have a total size of 1648 bits.

b) *Scalable to support large WSN:* The scalability of an access control mechanism is an essential design criterion for its implementation. For new node addition, many existing schemes in the literature necessitate the intervention of the Base Station. In Hyun-Sung and Sung-Woon (2009) and Haung (2009), BS transmits the public hash commitment after the node is added. Such intervention results in severe communication overhead, resulting in low scalability due to performance issues. A new node is added to the network based on the preloaded information stored during initialisation in the designed scheme. Furthermore, the scheme's design does not necessitate the intervention of the BS for new node addition thus enhancing scalability.

c) *Independent of time synchronisation issues:* To prevent the replay attack, the access control mechanism must include a mechanism to ensure the freshness of the messages exchanged. Zhou et al. (2007); Huang (2011), Das et al. (2013); Chatterjee et al. (2015) and Chatterjee and Roy (2018) used timestamps to ensure the freshness of messages. However, timestamps necessitate clock synchronisation between network nodes, causing

additional overhead on resource-constrained devices (Lasassmeh and Conrad, 2010). The proposed access control scheme associates node deployment with a deployment version. As discussed in Section 3, the deployment version is used to protect the freshness of a request thus making the designed scheme independent of time synchronisation issues.

Table 2 Symbols of critical operations

T_{EM}	Scalar Multiplication
T_E/ T_D	Encryption/Decryption
T_{HA}	Hash
T_{ECE}	ECC Encryption
T_{ECD}	ECC Decryption
T_{INV}	Modular Inverse
T_{PA}	Point Addition

Table 3 Parameters size

Parameter	*Size*
ND_I	16 bit
ND_J	16 bit
K_I	32 bit
Pu_{base}	320 bit
K_{base}	128 bit
D_{NI}	32 bit
CN_I	320 bit
SN_I	320 bit
$E_k[M]$: AES Encryption	128 bit
$H(.)$	160 bit

5 Security analysis

a) *Defence against False injection attack:* To prevent false data injection and eavesdropping of messages, security primitives of authentication and confidentiality are required. The bedrock of both the primitives is a shared key establishment between two communication parties. The proposed access control mechanism establishes a pair-wise key between each communicating neighbour. For nodes ND_I and ND_J, K_{IJ} is the pair-wise key established between them. K_{IJ} can provide encryption and authentication to thwart eavesdropping and false injection of packets with any lightweight symmetric technique. Similarly, other nodes can use the respective pair-wise keys established during authentication and key exchange handshake for secure communication with their neighbouring nodes.

b) *Defence against node capture attacks:* Let us consider a node ND_I, preloaded with the signature $(C_{NI}S_{NI})$, and having N neighbouring nodes. ND_I establishes a pair-wise key with all its adjacent nodes. Let K_{KEY} be the set of all the pair-wise keys, ND_I has established with its N neighbours where $K_{\text{KEY}} = \{K_{I1}, K_{I2}, K_{I3}, \ldots, K_{IJ}\}$ where $J = 1$ to N. Lets us assume that an adversary captures node ND_I. Besides other preloaded information, the attacker would have access to $(C_{NI}S_{NI})$ and K_{KEY}. However, due to the computational difficulty of the Elliptical Curve Discrete Lograthrimic Problem (Menezes, 2012; Hankerson et al., 2004), the secrets K_I and K_{base} cannot be extracted from $(C_{NI}S_{NI})$. Additionally, knowledge of the set K_{KEY} does not jeopardise the pair-wise keys of other network nodes. Thus, even if some nodes are captured and compromised, the proposed scheme's design does not jeopardise the network's overall security.

c) *Defence against Sybil attack:* A malicious node can claim multiple identities in a Sybil attack and cause severe disruption to the network operation. Let ND_I be the node whose identity an adversary wants to claim. The signature pair of the Node ND_I is given below: $(C_{NI}S_{NI})$

$$C_{NI} = K_I * K^1{}_{base} * G$$

$$S_{NI} = K_I * K^2{}_{base} * H(ND_I \,||\, D_{NI}) * G$$

Owing the Elliptical Curve Discrete Logarithmic Problem it is computationally infeasible for an attacker to derive K_I and K_{base} from $(C_{NI}S_{NI})$ (Menezes, 2012; Hankerson et al., 2004). Without knowledge of K_I and K_{base}, an adversary cannot create a forged signature pair with the correct deployment version and node identity. As a result, it becomes imperative that the proposed scheme is resistant to Sybil attack.

d) *Defence against Worm Hole Attack:* A wormhole attack comprises one or more malicious nodes in the network, which tunnel the data between them. The tunnel can be established between the new nodes or the old nodes. The designed scheme enables a node to join a network and be updated in neighbouring nodes' access control lists only after the deployed new node completes authentication and key exchange handshake with its neighbours. The deployment of a malicious wormhole node is secured because the signature of any node cannot be forged due to the computational difficulty of the Elliptical Curve Discrete Logarithm Problem (Menezes, 2012; Hankerson et al., 2004). As the proposed scheme is resistant against malicious node deployment, the wormhole attack is thwarted.

e) *Defence against man in the middle attack:* Let us assume that attacker A wants to execute a MITM between ND_I and ND_J. To do so, A Needs to falsify the signature

pair's $(C_{NI}S_{NI})$ and $(C_{NJ}S_{NJ})$, such that ND_J and ND_I accept them as legitimate signatures, respectively. For Attacker A, it is computationally infeasible to falsify the signature pair $(C_{NI}S_{NI})$ and $(C_{NJ}S_{NJ})$ due to Elliptical Curve Discrete Logarithm Problem (Menezes, 2012; Hankerson et al., 2004). Thus, MITM is thwarted in the proposed protocol.

f) *Defence against replay attack:* An old message can be replayed in the network in a replay attack to initiate an unsolicited and malicious authentication and key exchange handshake. In the proposed scheme, each node deployed in the network is associated with a deployment version. Let $(C_{NI},S_{NI},ND_I,D_{NI},PU_I)\,||\,H[C_{NI},S_{NI},ND_I,D_{NI},PU_I)]$ be an old access control request of Node ND_I. If the request is replayed later, the scheme design makes the request be rejected. In its access control list, each neighbouring node keeps the most recent deployment version of each Node. Let ND_J be the adjacent Node that receives the replayed request $(C_{NI},S_{NI},ND_I,D_{NI},PU_I)\,||\,H[C_{NI},S_{NI},ND_I,D_{NI},PU_I)]$. ND_J compares D_{NI} with D_{NI}^J. If $D_{NI} < D_{NI}^J$ or $D_{NI} = D_{NI}^J$ is true, then the request is rejected.

6 Formal security analysis using BAN logic

BAN Logic was proposed by Burrows et al. (1989) to validate the soundness and the correctness of the security protocol. This section employs BAN logic to assess the security of the designed scheme.

6.1 BAN model

BAN logic model primarily comprises syntax and postulates that are used for evaluating a security protocol. ND_I and ND_J denote the communicating principles, where (PU_I,K_I) and (PU_J,K_J) denote their public and private keys, respectively. The notations have been adapted from Islam and Biswas (2017). The BAN notations and BAN Postulates are tabulated in Table 4 and Table 5. Besides that, as derived in Buttyan et al. (1998), synthesis rules are further helpful in evaluating the correctness and soundness of security protocols. Synthesis rules are tabulated in Table 6. The notation $U \neg V$ implies V is derived and synthesised means from U.

6.2 BAN analysis

Let ND_I be the new Node, and ND_J be one of its neighbouring nodes with (K_I,P_{UI}) and (K_J,P_{UJ}) as their public/private key.

Table 4 Basic BAN logic postulates

Notation	*Description*
$ND_I \mid\equiv X$	Principal ND_I believes X.
$ND_I \Leftarrow X$	Principal ND_I receives the message X
$ND_I \mid\sim X$	ND_I sent the message X in past
$ND_I \mid\sim X$	ND_I sent the message X currently
$ND_I \mid\rightarrow X$	ND_I has control over X
$\#(X)$	X is fresh
$\xrightarrow{PU_I} ND_I$	PU_I is the public key of ND_I
$ND_I \xrightarrow{K_{IJ}} ND_J$	K_{IJ} is the key between ND_I and ND_J
$\{X\}_{KIJ}$	K_{IJ} is the key used to encrypt X.
$\frac{A}{B}$	if A is true, then B is true

Table 5 BAN postulates

Rule No.	*Name*	*Representation*
R1	Message-meaning rule	$\frac{ND_I \mid\equiv \xrightarrow{PU_J} ND_J, ND_I \Leftarrow \{X\}_{KIJ}}{ND_I \mid\equiv ND_I \mid\sim X}$
R2	Nonce verification rule	$\frac{ND_I \mid\equiv \#(X), ND_I \equiv ND_J \mid\sim X}{ND_I \mid\equiv ND_J \mid\equiv X}$
R3	Jurisdiction rule	$\frac{ND_I \mid\rightarrow X,\ ND_I \mid\equiv ND_J \mid\equiv X}{ND_I \mid\equiv X}$
R4	Seeing rule	$\frac{ND_I \Leftarrow X, ND_I \Leftarrow Y}{ND_I \Leftarrow (X,Y)}$
R5	Belief rule	$\frac{ND_I \mid\equiv X, ND_I \mid\equiv Y}{ND_I \mid\equiv (X,Y)}$
R6	Freshness rule	$\frac{ND_I \mid\equiv \#(X)}{ND_I \mid\equiv \#(X,Y)}$
R7	Session key rule	$\frac{ND_I \mid\equiv \#(K_U), ND_I \mid\equiv ND_J \mid\equiv X}{ND_I \mid\equiv ND_I \xrightarrow{K_U} ND_J}$

Table 6 Synthesis rule's

Rule No.	*Synthesis rule*
SU1	$ND_I \Leftarrow X \neg ND_I \Leftarrow (X,Y)$
SU2	$ND_I \mid\equiv ND_J \mid\sim X \neg ND_I \mid\equiv ND_J \mid\sim (X,Y)$
SU3	$ND_I \mid\equiv ND_J \mid\sim (X,Y) \neg ND_I \mid\equiv ND_J \mid\sim X$
SU4	$ND_I \mid\equiv ND_J \mid\sim X \neg ND_I \mid\equiv \#(X)$

6.2.1 Assumptions

A1) $ND_I \mid\equiv \xrightarrow{PU_I} ND_I$ A2) $ND_J \mid\equiv \xrightarrow{PU_I} ND_I$

A3) $ND_J \mid\equiv \xrightarrow{PU_J} ND_J$ A4) $ND_I \mid\equiv \xrightarrow{PU_J} ND_J$

A5) $ND_I \mid\equiv \#(D_{NI})$ A6) $ND_J \mid\equiv \#(D_{NJ})$

A7) $ND_J \mid\equiv ND_I \mid\rightarrow D_{NI}$ A8) $ND_I \mid\equiv ND_J \mid\rightarrow D_{NJ}$

6.2.2 Idealised messages

The idealised form of the two messages exchanges between ND_I and ND_J are listed as below:

$$ND_I \rightarrow ND_J; \{C_{NI}\}K_I, \{S_{NI}\}K_I (MS1)$$
$$ND_J \rightarrow ND_I; \{C_{NJ}\}K_J, \{S_{NJ}\}K_J (MS2)$$

6.2.3 Goals to be achieved

To prove that the proper authentication mechanism is established in the proposed scheme following goals must be achieved:

G1) $ND_J \mid\equiv ND_I \xrightarrow{K_{IJ}} ND_J$ G2)
$ND_J \mid\equiv ND_I \mid\equiv ND_I \xrightarrow{K_{IJ}} ND_J$

G3) $ND_I \mid\equiv ND_I \xrightarrow{K_{IJ}} ND_J$ G4)
$ND_I \mid\equiv ND_J \mid\equiv ND_I \xrightarrow{K_{IJ}} ND_J$

6.2.4 BAN verification

The verification steps are given as under:

From (MS1):

B1) $ND_I \mid\equiv \{CN_I\}K_I, \{SN_I\}K_I$

B2) $ND_J \Leftarrow \{CN_I\}K_I$

B3) $ND_J \Leftarrow \{SN_I\}K_I$

From (B3) and (A2) and applying (R1), we get:

B4) $ND_J \mid\equiv ND_I \mid\sim \{SN_I\}K_I$

DN_I is a part of SN_I, thus as per (A5) and (R6), we get:

B5) $ND_I \mid\equiv \#(SN_I)$

From B4 and B5, we get:

B6) $ND_J \mid\equiv ND_I \mid\sim SN_I$

From (B6) and the (SU4), we get:

B7) $ND_J \mid\equiv \#(SN_I)$

From (B4) and (B7) on applying (R2), we get:

B8) $ND_J \mid\equiv ND_I \mid\equiv SN_I$

D_{NI} is the part of SN_I; thus, on applying (R5), we get:

B9) $ND_J \mid\equiv ND_I \mid\equiv D_{NI}$

Now, as per (A7) and (B9) and applying (R3), we get:

B10) $ND_J \mid\equiv D_{NI}$

From (SU3) and (B4), we get:

B11) $ND_J \mid\equiv ND_I \mid\sim D_{NI}$

From (A5) and (B11), we get:

B12) $ND_J \mid\equiv ND_I \mid \mid\sim D_{NI}$

As per (SU4) and (B12), we get:

B13) $ND_J \mid\equiv \#(D_{NI})$

D_{NI} is a part of Session Key (K_{IJ}) thus, as per (R6) we get:

B14) $ND_J \mid\equiv \#(K_{IJ})$

From (B14) & (B9) and (R7)

B15) $ND_J \mid\equiv ND_I \xrightarrow{K_{IJ}} ND_J$

Owing the symmetry of the protocol, ND_I believes that ND_J is bound to derive the same believe

B16) $ND_J \mid\equiv ND_I \mid\equiv ND_I \xrightarrow{K_{IJ}} ND_J$

From (MS2) we infer that:

B17) $ND_J \mid\equiv \{CN_J\}K_J, \{SN_J\}K_J$

B18) $ND_I \Leftarrow \{CN_J\}K_J$

B19) $ND_I \Leftarrow \{SN_J\}K_J$

From (B19) and (A4) and applying (R1), we get:

B20) $ND_I \mid\equiv ND_J \mid\sim \{SN_J\}K_J$

D_{NJ} is a part of SN_J, thus as per (A5) and (R6), we get:

B21) $ND_J \mid\equiv \#(SN_J)$

From B20 and B21, we get:

B22) $ND_I \mid\equiv ND_J \mid \mid\sim SN_J$

From (B22) and (SU4), we get:

B23) $ND_I \mid\equiv \#(SN_J)$

From (B20) and (B23) on applying (R2), we get:

B24) $ND_I \mid\equiv ND_J \mid\equiv SN_J$

D_{NJ} is the part of SN_J; thus, on applying (R5), we get :

B25) $ND_I \mid\equiv ND_J \mid\equiv SN_J$

Now, as per (A8) and (B25) and applying (R3), we get:

B26) $ND_I \mid \equiv SN_J$

From (SU3) and (B20), we get:

B27) $ND_I \mid \equiv ND_J \mid \sim SN_J$

From (A6) and (B27), we get:

B28) $ND_I \mid \equiv ND_J \mid \sim D_{NJ}$

As per (SU4) and (B28), we get:

B29) $ND_I \mid \equiv \#(D_{NJ})$

SN_J is a part of (SK_{IJ}) thus, as per (R6) we get:

B30) $ND_I \mid \equiv \#(K_{IJ})$

From (B30) & (B25) and on applying (R7)

B31) $ND_I \mid \equiv NS_I \xrightarrow{K_{IJ}} ND_J$

Owing the symmetry of the protocol,

B32) $ND_J \mid \equiv ND_I \mid \equiv ND_I \xrightarrow{K_{IJ}} ND_J$

7 Formal analysis using the Random Oracle model

This section presents formal security proof using the Random Oracle model to prove that the proposed scheme is resistant to Wormhole attack, Sybil attack and Node cloning attack. The procedure followed in the proof is based on Chatterjee and Roy (2018).

Theorem: *The Scheme is resilient to Wormhole attack, Sybil Attack, and Node cloning attack under the ECDLP assumption.*

Proof: Let $U \in E(a,b)$ such that $V = k.U$ where $k \in Z_P$. According to ECDLP, finding a $r \in Z_P$ where $r \neq k$ and $V = r.U$ is computationally infeasible. Consider the following two distributions to determine the advantage of any probabilistic polynomial-time distinguisher in solving ECDLP on a curve $E(a,b)$.

$$\blacktriangle_{re} = \{k \in Z_P,\ P = U, Q = V = (k.U), R = k : (P,Q,R)\}$$

$$\blacktriangle_{ra} = \{k, r \in Z_P, P = U, Q = V = (k.U), R = r : (P,Q,R)\}$$

To solve the ECDLP, the advantage of any probabilistic polynomial-time distinguisher *D* with a binary output (0/1) is given as:

$$ADVT_D^{ECDLP} = \left| P\left[(P,Q,R) \leftarrow \blacktriangle_{re} : D(P,Q,R) = 1\right] - P\left[(P,Q,R) \leftarrow \blacktriangle_{ra} : D(P,Q,R) = 1\right] \right|$$

where $P[]$ is a random probability over *k* and *r*, *D* is considered to be a (T,e) distinguisher for $E(a,b)$ if *D* runs in a time *T* such that:

$$ADVT_D^{ECDLP}(T) \geq e$$

However, as per ECDLP, there exists no polynomial-time distinguisher (T,e) for a curve $E(a,b)$. The formal proof is further based on the method of contradiction, as discussed in Das et al. (2013) and Chuang and Tseng (2010).

Let us assume that adversary A can extract the private key K_{base} of the BS. We further believe that an adversary can also determine the private key K_I of ND_I. Based on these assumptions, adversary A would compute a malicious signature $(C_{MI} S_{MI})$ of an ND_I with a proper deployment version. Thus we define the following oracles for the adversary:

1) *Reveal BaseKey*: This query outputs the private key K_{base} of the BS using $E(a,b)$ and $PU_{base} = K_{base}.G$ as input.

2) *RevealNodeKey*: This query outputs the private key K_I of ND_I using $E(a,b)$ and $PU_I = K_I.G$ as input

3) *CreateSignaturePair*: This query allows an attacker to generate a malicious node's signature pair $(C_{MI} S_{MI})$.

Adversary runs the experiment $EXP_{ACS}^{E(a,b)}$ as shown in Figure 4.

The success of the experiment is defined as:

$$Success_{ACS}^{ECDLP} = 2P\left[EXP_{ACS}^{E(a,b)} = 1\right] - 1$$

Accordingly, the advantage is defined as :

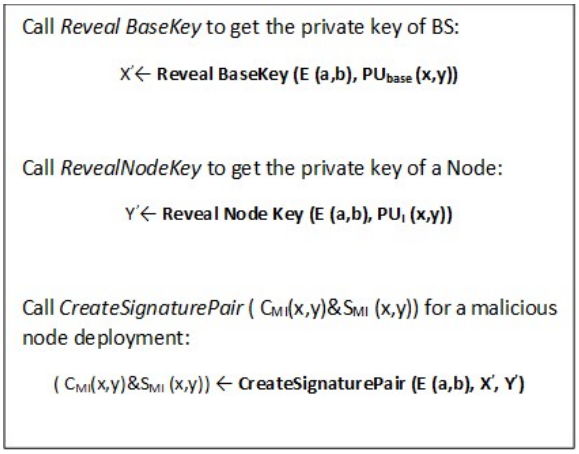
$$ADVT_{ACS}^{ECDLP}(t, Q_B, Q_N, Q_S) = \text{Max}_A \left\{Success_{ACS}^{ECDLP}\right\}$$

where in maximum is taken over all execution *t*, Q_B is the number of queries to the *Reveal BaseKey*, Q_N is the number of queries to the *RevealNodeKey*, Q_S is the number of queries the *CreateSignaturePair*. The proposed protocol would be secure against wormhole attack, Sybil Attack and Node cloning attack if:

$$ADV_{ACS}^{ECDLP}(t, Q_B, Q_N, Q_S) \leq e$$

where $e > 0$

Figure 4 Experiment $EXP_{ACS}^{E(a,b)}$ run by the adversary



Based on the experiment shown in Figure 4, an adversary can extract the private key of BS and a Node. Subsequently, the adversary generates the signature pair C_{MI} and S_{MI} through the additive splitting of the BS private key with a proper deployment version. However, as per ECDLP, extracting the private key of the base station and node is a computationally infeasible problem. Thus, we can conclude that:

$$ADV_{ACS}^{ECDLP}\left(t, Q_{B,} Q_N, Q_S\right) \le e$$

where $e > 0$

Hence, the proposed scheme provides a strong resilience to Wormhole attack, Sybil attack and Node cloning attack.

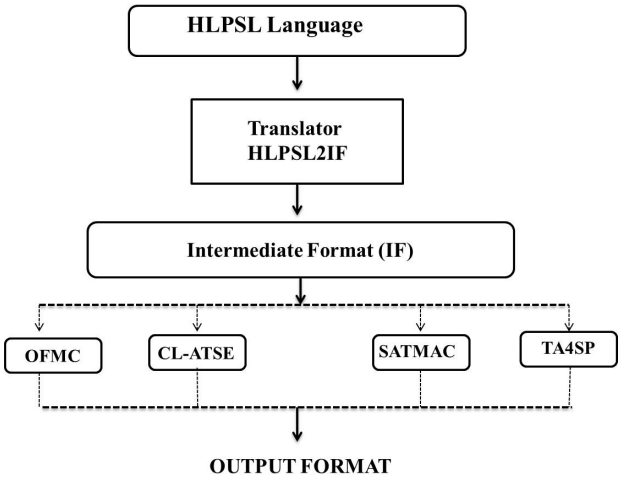
8 Simulation for formal security validation

In this section, simulations on AVISPA and Scyther tools have been carried out to verify and validate the security strength of the proposed protocol.

8.1 AVISPA simulation

AVISPA is used (Clarke et al., 1998) to validate and verify whether the desired security goals of the security protocol are achieved. The schematic of the tool is given in Figure 5. The HLPSL programming language is used to model the security protocol. A translator, HLPSL2IF, converts the HLPSL Model into IF format. The four backends then evaluate the IF Format to provide protocol falsification and verification of the desired security protocol. The description of the various backend's is given in Table 7. The HLPSL models the communicating parties as roles that are played by designated agents. The definition of roles models the communication pattern in terms of states and transitions. The role session serves as a model for the session's composition and initiation. The role environment is made up of multiple sessions that run in parallel and intruder knowledge and other parameters that define the environment. More information on AVISPA is given in Armando et al. (2005).

Figure 5 AVSIPA architecture



The Complete HLPSL script of the protocol is given in Appendix A. The authentication and the key exchange between the ND_I and ND_J are modelled by defining their corresponding HLPSL roles. The roleNewNode is played by the agent ND_I. The RCV (start) in the state 0 of the roleNewNode initiates the simulation. On receiving the start, the agent ND_I sends the broadcast: $\left(C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I\right) \| H\left[C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I\right)\right]$ using the SND() operation. SND and RCV are defined as a channel (dy). Channel (dy) defines the Dolev and Yoa threat model in which the communication channel is completely insecure. In state 0, K_I is specified to be a secrecy goal identified by protocol_id type Key_KI. The roleNewNode in state 1, on receiving the response $\left(C_{NJ} S_{NJ}, ND_J, D_{NJ}, PU_J\right) \| H\left[C_{NJ} S_{NJ}, ND_J, D_{NJ}, PU_J\right] \|$ $\| E_{Kij}\left[R_N\right] \| H\left[R_N\right]$ using the RCV() from neighbouring node NDJ, the conjunction, request (NDI,NDJ, NDJ_NDI, RN') is validated. Request (NDI, NDJ, NDJ_NDI, RN') is a strong authentication where agent NDJ witnesses the RN' for NDI and is identified by NDJ_NDI in the goal section. The agent NDJ plays the roleNeighNode. On receiving $\left(C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I\right) \| H\left[C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I\right)\right]$ using RCV(), the role NeighNode played by agent NDJ sends $\left(C_{NJ} S_{NJ}, ND_J, D_{NJ}, PU_J\right) \| H\left[C_{NJ} S_{NJ}, ND_J, D_{NJ}, PU_J\right]$ $\left\|E_{Kij}\left[R_N\right]\right\| H\left[R_N\right]$ using SND() operation. The K_J is specified to be a secrecy goal identified by Key_KJ. The witness (NDJ, NDI, NDJ_NDI, RN') demands a weak authentication of NDI by NDJ where NDJ witnesses the information given by NDI, i.e. RN.' A session is a composing role instantiating one or more basic roles. The composed role doesn't have a transition section. $\wedge$ is used to indicate the basic role that runs in parallel. The security goals for the validation of the proposed scheme are tabulated in Table 8.

Table 7 Description of AVISPA backends

Backend	*Description*
OFMC	OFMC stands for On-the-Fly-Model-Checker. It provides techniques to explore the state space in a demand-driven way.
CL-AtSe	CL-AtSe stands for constraint attack-based logic searcher. It converts transition relation in a set of constraints to analyse attacks
SATMC	SATMC stands for SAT-based Model-Checker. It Builds prepositional formulae for SAT solver.
TA4SP	TA4SP stands for TREE automata based on automatic approximations for analysis of security protocols. By employing regular tree languages, it estimates the intruder's knowledge.

Table 8 AVSIPA goals

Goals	*Description*
secrecy_of Key_KI	K_I must remain secret to ND_I
secrecy_of Key_KJ	K_J must remain secret to ND_J
authentication_on NDI_NDJ	ND_I is correct in believing that ND_J in the current session
authentication_on NDJ_NDI	ND_J is correct in believing that ND_I in the current session

The HLPSL code was simulated using SPAN, simulation animator of AVISPA. Figure 6 depicts the corresponding message sequence chart on SPAN, which shows 02 messages being exchanged. The proposed protocol's HLPSL model has been validated on the OFMC backend. The OFMC backend uses symbolic techniques to generate state representations on the fly. OFMC detects attacks quickly in a limited number of sessions. To validate the replay attack in the proposed scheme, the backend searches for a passive intruder. The MITM is countered by inserting an active intruder *i*. Figure 7 depicts the simulation results on the OFMC backend. The result summary section indicates that the scheme is SAFE and that the desired security goals have been met. Thus, the AVISPA verification indicates that the protocol is resistant to a MITM attack. The search time is 0.25 seconds, and the number of nodes visited is 24, with a depth of 60.

Figure 6 SPAN simulation of the proposed access protocol

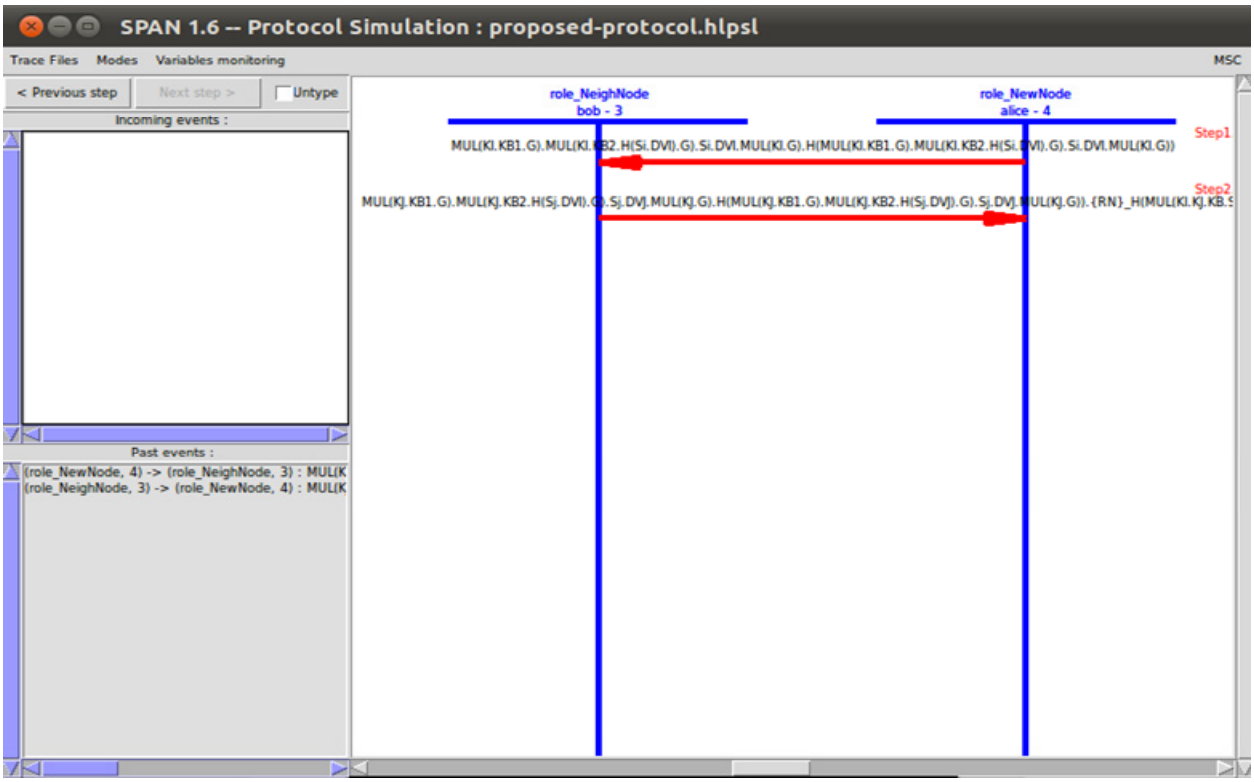
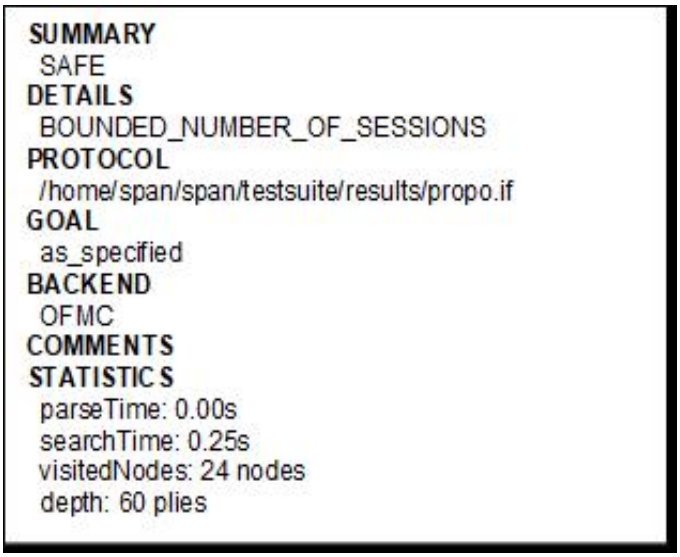


Figure 7 AVISPA verification results



8.2 Scyther simulation

Scyther (Cremers, 2008) is a framework for formal verification of security protocols with an unbounded no of sessions. Scyther Protocol Description Language (SPDL) is used to model a security protocol that will be verified in Scyther. Scyther is discussed in further detail in (Cremers, 2008). The SPDL model of the designed scheme is given in Appendix-B. The SPDL modelling of the proposed scheme comprises two roles: (1) role I (2) role R. role I model the communication of an ND_I, and role R models the node ND_J. The private key of BS is declared as secret, hence not accessible within the session. The send function is used to send $(C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I) \| H[C_{NI}, S_{NI}, ND_I, D_{NI}, PU_I)]$ to role R. Subsequently recv function is used to receive $(C_{NJ} S_{NJ}, ND_J, D_{NJ}, PU_J) \| H[C_{NJ} S_{NJ}, ND_J, D_{NJ}, PU_J] \| E_{Kij}[R_N] \| H[R_N]$ using recv function. The security goals desired to be verified for analysing the proposed protocol are shown in Table 9. Figure 8 depicts the Scyther simulation parameters. Figure 9 illustrates that protocol is safe against various attacks.

Figure 8 Simulation parameters in Scyther

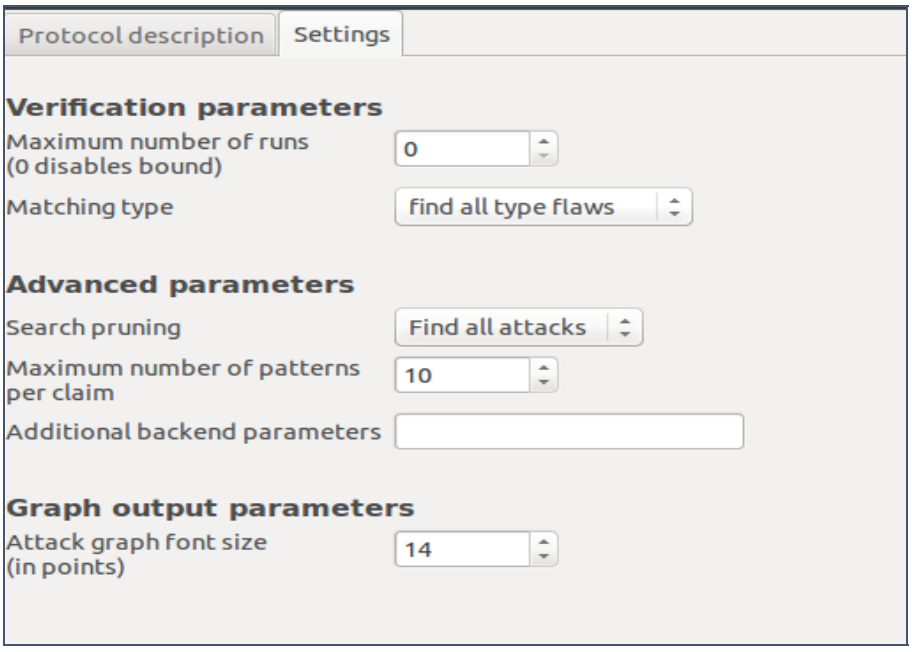
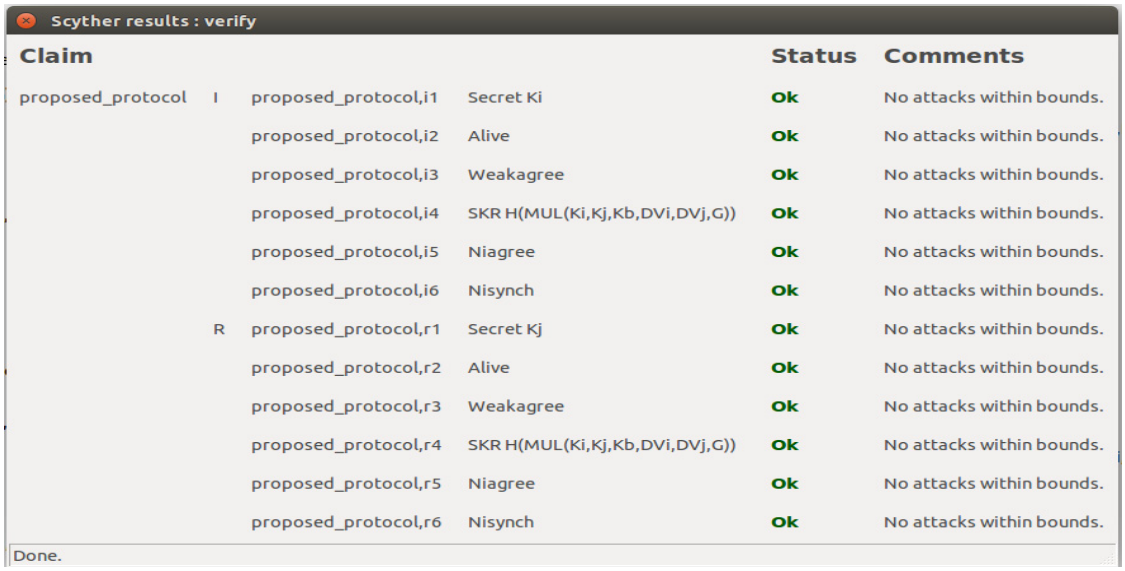


Table 9 Scyther security claims

Claim	*Description*
Secret KI	The confidentiality of K_I must be maintained and remain accessible to ND_I only.
Secret KJ	The confidentiality of K_J must be maintained and remain accessible to ND_J only.
SKR: H(MUL (KI, KJ, Kb, DVI, DVJ G))	SKR: H(MUL (KI, KJ, Kb, DVI, DVJ G)) is the key established between ND_I and ND_J
Alive	Alive claim indicates that communicative parties, ND_I and ND_J, are exchanging messages with an intending partner.
Weak_Agree	Weak_Agree claim indicates that communicative parties, ND_I and ND_J, are exchanging messages with an intending partner recently.
NiAgree	NiAgree claim indicates that communicative parties, ND_I and ND_J, are agreeing on the messages exchanged
Ni-SYNCH	The Ni-SYNCH claim indicates that communicative parties, ND_I and ND_J, are synchronised

Figure 9 Scyther verification results



9 Performance comparison with other schemes

To compare the computational overhead of various exiting schemes to the proposed access control scheme, Table 10 summarises the time complexity of various critical operations in terms of T_{MMUL} (Modular Multiplication), as specified in Wen et al. (2011). The time required for various critical operations on the MicaZ mote is indicated in Table 11 based on experimentation in Iqbal and Mir (2020a). The overhead comparison in terms of computation, communication, memory, and estimated time is taken for critical operations is shown in Table 12. Zhou et al. (2007) exhibited the highest computational time complexity, while Chatterjee et al. (2015) exhibit the least. Subsequently, Zhou et al. (2007) also have the highest communication and memory overhead. The lowest computational time taken for critical operations is Chatterjee et al. (2015) and the highest time is taken by Zhou et al. (2007). The criteria used to categorise the overhead in terms of High, Medium and Low, as indicated in Chatterjee and Roy (2018), is shown in Table 13. Table 14 compares the proposed scheme to existing schemes in general. Chatterjee et al. (2015) low overhead in terms of communication, computation and memory. It also supports all security requirements from SE-R^1 to SE-R^6. However, the scheme is neither scalable nor is independent of time synchronisation, thus not making it practical for WSN. The (Zhou et al., 2007) scheme is scalable but has high computational, communication, and memory overhead. Haung (2009) does not supported SE-R^1, SE-R^2, SE-R^5 and SE-R^6 and is not scalable. Chatterjee and Roy (2018) supported all security requirements and are scalable. Additionally, the scheme has low overheads but is dependent on clock synchronisation between network nodes. The formal validation is provided only by Das et al. (2013); Chatterjee et al. (2015); Chatterjee and Roy (2018), and the proposed scheme. Furthermore, it can be elucidated that the proposed scheme is the only one that does not rely on clock synchronisation between network nodes. From Table 14, it can be depicted that in terms of functionality, security strength and efficiency, the proposed scheme is better than the existing scheme's, thus making it viable for practical usage in the wireless sensor network.

Table 10 Time complexities in terms of T_{MMUL}

Symbol	*Time complexity in T_{MM}*
T_{EM}	1200 T_{MMUL}
T_E/ T_D	3 T_{MMUL}
T_{HA}	0.36 T_{MMUL}
T_{ECE}	2405 T_{MMUL}
T_{ECD}	1205 T_{MMUL}
T_{INV}	0.15 T_{MMUL}
T_{PA}	5 T_{MMUL}

Table 11 Time taken on MicaZ mote

Symbol	*Time*
T_{EM}	2.82
T_{INV}	0.14
T_{HA}	.0091
T_{ECE}	3.9
T_{ECD}	2.6
T_E/ T_D	0.000029
T_{PA}	0.16

Table 12 Comparison of computational, communication and memory overhead

Scheme	*Computational overhead*	*Communication overhead (Bits)*	*Memory overhead (Bytes)*	*Estimated time required for critical operations (Seconds)*
Zhou et al. (2007)	$3T_{EM} + T_{INV} + T_{HA} + 2T_{ECE}/T_{ECDC} \approx 7213\ T_{MMUL}$	9152	228	16.40
Huang (2009)	$2T_{EM} + 4T_{HA} \approx 2401\ T_{MMUL}$	3328 + 160 *n	162	5.67
Hyun-Sung and Sung-Woon (2009)	$2T_{EM} + 9T_{HA} \approx 2409\ T_{MMUL}$	3328 + 512 *n	202	5.71
Huang (2011)	$5T_{EM} + 4T_{HA} \approx 6001\ T_{MMUL}$	3456	206	14.13
Das et al. (2013)	$4T_{EM} + 4T_{HA} + T_{INV} + T_E/T_D \approx 4805\ T_{MMUL}$	4224	195	11.45
Chatterjee et al. (2015)	$8T_{HA} + T_E/T_D$	1800	112	0.072
Chatterjee and Roy (2018)	$2T_{EM} + 5T_{HA} + T_E/T_D + T_{PA} \approx 2408\ T_{MMUL}$	4288	208	5.84
Proposed Scheme	$2T_{EM} + 3T_{HA} + T_{PA} + T_E/T_D + T_{INV} \approx 2409\ T_{MMUL}$	5248	206	5.96

Table 13 Criteria for overhead categorisation

Overhead	*High*	*Medium*	*Low*
Computation	>4000 T_{MMUL}	2000 to 4000 T_{MMUL}	0 to 2000 T_{MMUL}
Communication	>9000 Bits	6000 to 9000 Bits	0 to 6000 Bits
Memory	>220 Bytes	150 to 200 Bytes	0 to 150 Bytes

Table 14 Overall comparison

Scheme	*SE-R*^{1}	*SE-R*^{2}	*SE-R*^{3}	*SE-R*^{4}	*SE-R*^{5}	*SE-R*^{6}	*(FU-R*^{1}*)*	*(FU-R*^{1}*)*	*(FU-R*^{1}*)*	*FU-R*^{2}	*FU-R*^{3}	*Formal validation?*
Zhou et al. (2007)	✓	✓	✓	✓	✓	✓	High	High	High	✓	✗	✗
Huang (2009)	✗	✗	✓	✓	✗	✗	High	Medium	Medium	✗	✓	✗
Hyun-Sung and Sung-Woon (2009)	✗	✗	✓	✓	✗	✗	High	Medium	Medium	✗	✓	✗
Huang (2011)	✗	✗	✓	✓	✗	✗	Low	High	Medium	✓	✗	✗
Das et al. (2013)	✓	✓	✓	✓	✓	✓	Low	High	Medium	✓	✗	✓
Chatterjee et al. (2015)	✓	✓	✓	✓	✓	✓	Low	Low	Low	✗	✗	✓
Chatterjee and Roy (2018)	✓	✓	✓	✓	✓	✓	Low	Medium	Medium	✓	✗	✓
Proposed Scheme	✓	✓	✓	✓	✓	✓	Low	Medium	Medium	✓	✓	✓

Notes: SE-R^1: Resistant against eavesdropping and false injection attack; SE-R^2: Resistant against node capture attack; SE-R^3: Resistant against Sybil attack; SE-R^4: Resistant against wormhole attack; SE-R^5: Resistant against Man in the Middle attack; SE-R^6: Resistant against replay attack; FU-R^1: Computation, Communication and Memory Efficiency; FU-R^2: Must be scalable to Large WSN; FU-R^3: Independent of Time Synchronisation issues.

10 Practical experimentation

Practical experimentation based on TinyOS operating using the TinyECC library has been carried out to perform TOSSIM simulation and MICAz test-bed implementation. TinyOS is an asynchronous and component-based IoT/WSN operating system. TinyOS is component-based and supports a modular architecture. The event-driven architecture is supported using asynchronous methods or services. Each service provided by a component is implemented using interfaces. Interfaces act as a contract between a component providing the service and the component implementing the service. Components are of two

different kinds: (1) Configuration and (2) Modules. A configuration is a component in which various components used in the application are wired with the application module using interfaces. NesC is the language used in TinyOS. The proposed scheme has been implemented on TinyOS using the TinyECC library. To carry out the practical implementation, the designed scheme has been modeled in NesC, and the corresponding component graph is shown in Figure 10. The descriptions of the various NesC components used in the component graph are given in Table 15. The description of various NesC interfaces used in the proposed protocol is shown in Table 16.

Figure 10 NesC component graph

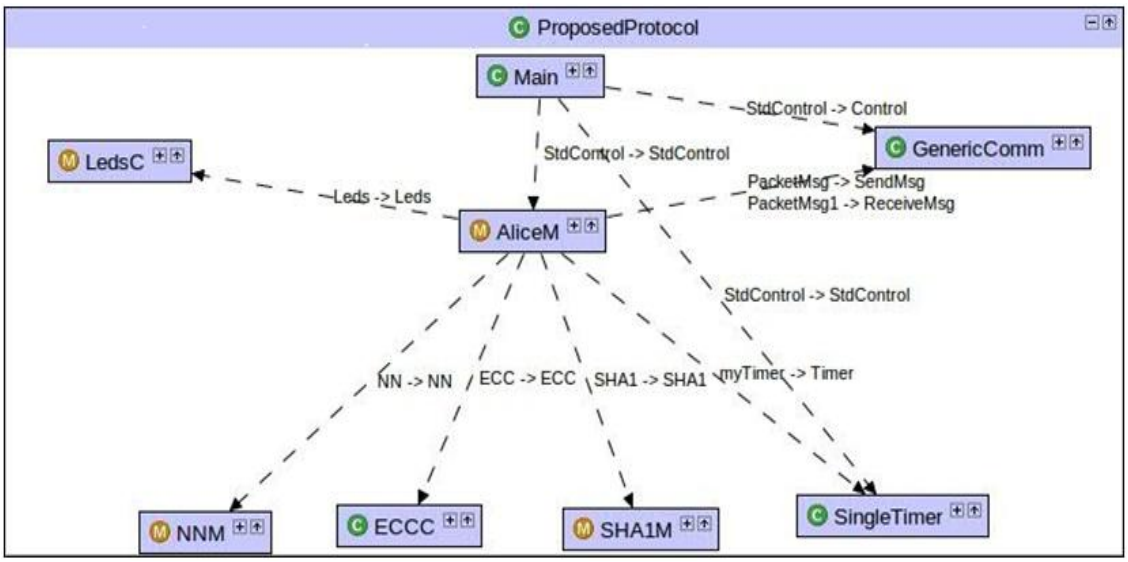


Table 15 NesC components used for TinyOS implementation

Module name	*Description*
Main	It is a critical component in TinyOS that initiates execution.
LedC	Provides an implementation for controlling LEDs.
GennericComm	Provides an asynchronous implementation of generic communication operations such as to send and receive.
NNM	It is part of the TinyECC library and implements a variety of number theory operations.
ECCC	It implements a variety of ECC operations, including point multiplication, addition and scalar multiplication.
SHA1M	It implements 160-bit SHA 1 hash.
Single Timer	Timing control implementation is provided.
AliceM	Provides the application logic for the proposed protocol.

Table 16 NesC interface's used for TinyOS implementation

Interface name	*Description*
StdControl	Declares the contract of standard control operations, which include start, stop, initialise of the application
LedC	Declares the contract of led operations, which include On and Off
NNM	Declares the contract of Number theory operations, which include multiplication inverse, addition, etc.
ECC	Declares the contract of elliptical curve cryptography operations, which include scalar multiplication, point addition, etc.
SHA1	Declares the contract, which includes command and events for the creation of SHA1 160 bit digest
Timer	Declares the contract for timing operations, which include staring the timer and handling timer events
Send/Receive	Declares the contract for asynchronous communication, which includes sending, send done and receive

Practical experimentation on TinyOS and using TinyECC includes the following:

1) Simulation on TOSSIM

2) Test-Bed implementation using MicaZ

10.1 Simulation on TOSSIM

TOSSIM is an inbuilt simulator of TinyOS, which simulates IoT/WSN applications realistically by taking into account the real-time noise and signal propagation models. TOSSIM also has a java based visualisation environment called TinyVIZ. To provide energy estimation, TOSSIM provides an energy plugin called PowerTOSSIM (Iqbal and Mir, 2021c). More details on PowerTOSSIM can be found at Shnayder et al. (2004). The NesC application was simulated on TOSSIM using the simulation parameters shown in Table 17. The energy model employed in the simulation is a Mica2 model, as shown in Figure 10. TinyVIZ snapshot depicting the energy consumed per Node is shown in Figure 11.

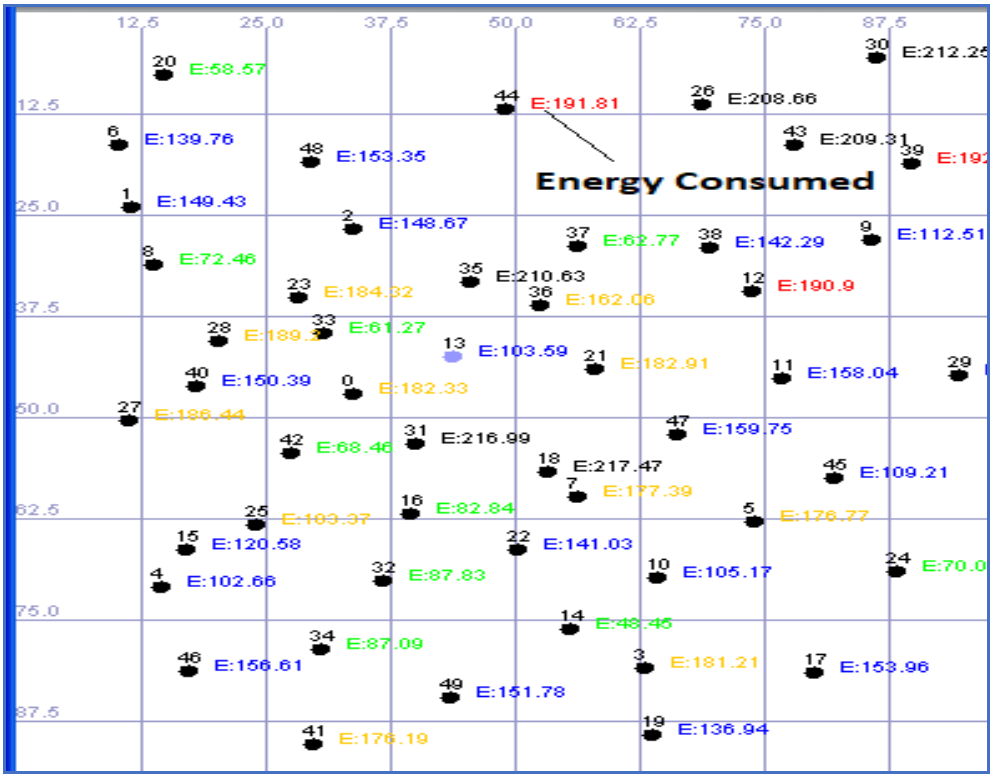
Table 17 Simulator parameters

Sl. No.	*Parameter*	*Description*
1	Deployment	Random
2	Distance between nodes	5 ft
3	Curve	Secp160r1
4	Communication model	Lossy Model
5	Communication channel	13
6	Active message type	19
7	Number of nodes	50

Figure 10 MICA2 model

CPU		Radio		LED/Sensor Board	
Active	8.0 mA	Rx	7.0 mA	Led's	6.2 mA
Idle	3.2 mA	Tx(-20 dBm)	3.7 mA	Sensor Board	0.7 mA
ADC Noise Reduce	1.0 mA	Tx(-19 dBm)	5.2 mA	**EEPROM**	
Power Down	103 µA	Tx(-15 dBm)	5.4 mA		
Power Save	110 µA	Tx(-8 dBm)	6.5 mA	Read	6.2mA
Stand By	216 µA	Tx(-5 dBm)	7.1 mA	Read Time	565 µs
Extended Standby	223 µA	Tx(0 dBm)	8.5 mA	Write	18.4 mA
Internal Oscillator	0.93 µA	Tx(+4 dBm)	11.6 mA	Write Time	12.9 ms

Figure 11 TinyVIZ simulation



10.2 Test-Bed implementation

The scheme was tested on a Test Bed consisting of five MicaZ motes. The proposed protocol was implemented in nesC on four Micaz Nodes with the Node IDs 1, 2, 3, 4 using the MIB520 (Moteworks, 2013) programmer board and deployed with $D_N[1, 2, 3, 4] = 1$. The proposed protocol's total ROM and RAM consumption per node including TinyOS scheduler and other system components is depicted in Figure 12. New Node N_5 with the Id 5 has been further deployed with the $D_{N5} = 2$. To check the ACL entries in a node, an ACL broadcast packet was created within the payload of the TinyOS message. The ACL broadcast packet is shown in Figure 13. The ACL packets were sniffed using a Perytron analyser (Perytons Protocol Analyzer, 2014). The ACL broadcast packets from New Node N_5 captured by Peryton depicting the ACL entries of the neighbouring nodes (N_1 N_2, N_3, N_4) are shown in Table 18.

Figure 12 Memory on the MicaZ node

```
    compiled Alice to build/micaz/main.exe
          15630 bytes in ROM
           1277 bytes in RAM
avr-objcopy --output-target=srec build/micaz/main.exe build/micaz/main.srec
avr-objcopy --output-target=ihex build/micaz/main.exe build/micaz/main.ihex
    writing TOS image
```

Figure 13 ACL broadcast packet

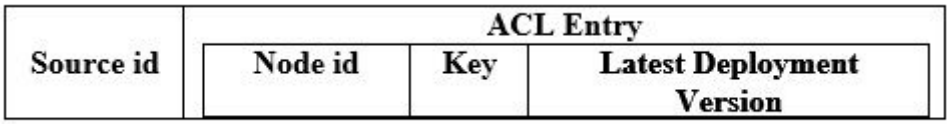


Table 18 Messages captured using Perytron

ACL Entry	*Node Id*	*Perytron packet*
N_5 and N_1	N_5	Msg 6 | Info | Mac | NWL | Payload 20 | source id | Data | ACL Entry | 7D050101 0D0C0216 3379E4D7 B17DECC7 913D6CC2
	N_1	Msg 10 | Info | Mac | NWL | Payload 20 | Source Id | Data | ACL Entry | 7D010502 0D0C0216 3379E4D7 B17DECC7 913D6CC2
N_5 and N_2	N_5	Msg 7 | Info | Mac | NWL | Payload 20 | Source ID | Data | ACL Entry | 7D050201 8D9027DF 868AA00A 1A1FB4A4 2B202DF0
	N_2	Msg 12 | Info | Mac | NWL | Payload 20 | Source Id | Data | ACL Entry | 7D020502 8D9027DF 868AA00A 1A1FB4A4 2B202DF0
N_5 and N_3	N_5	Msg 8 | Info | Mac | NWL | Payload 20 | Source ID | Data | ACL Entry | 7D050301 7C281E74 13ADC00D A47CCB7A 756D8362
	N_3	Msg 17 | Info | Mac | NWL | Payload 20 | Source ID | Data | ACL Entry | 7D030502 7C281E74 13ADC00D A47CCB7A 756D8362
N_5 and N_4	N_5	Msg 9 | Info | Mac | NWL | Payload 20 | Source ID | Data | ACL Entry | 7D050401 0DC6D3CE 678AA5B9 B18DAC67 A19D4C39
	N_4	Msg 21 | Info | Mac | NWL | Payload 20 | Source ID | Data | ACL Entry | 7D040502 0DC6D3CE 678AA5B9 B18DAC67 A19D4C39

11 Conclusion

New node deployment is an essential requirement in WSN. However, to prevent a malicious node deployment, an access control scheme needs to be enforced. In this paper, a design of an efficient new node access control scheme based on ECC is presented. The designed scheme applies to large WSNs and does not require clock synchronisation between network nodes. The scheme has been evaluated using provable security techniques which includes the BAN logic and Random Oracle Model. According to the evaluation, the scheme is secure and resistant to a variety of attacks. Additionally, AVISPA and Scyther simulation demonstrates that the scheme is resistant to a variety of active and passive attacks. Experiments on TinyOS with the TinyECC library were conducted to perform a detailed simulation and test-bed implementation. The simulation was conducted using the TOSSIM simulator, and the energy requirements were analysed using the PowerTOSSIM plugin. The test-bed implementation has been done on MicaZ motes to highlight the working details of the proposed scheme on practical WSN motes.

References

Abdmeziem, M., Tandjaoui, D. and Romdhani, I. (2015) 'Architecting the internet of things: state of the art', *Robots and Sensor Clouds*, pp.55–75.

Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuellar, J., Drielsma, P., Heám, P., Kouchnarenko, O., Mantovani, J., Mödersheim, S., von Oheimb, D., Rusinowitch, M., Santiago, J., Turuani, M., Viganò, L. and Vigneron, L. (2005) 'The AVISPA tool for the automated validation of internet security protocols and applications', *Computer Aided Verification*, pp.281–285.

Bukhari, S.H.R., Siraj, S. and Rehmani, M.H. (2018) 'Wireless sensor networks in smart cities: applications of channel bonding to meet data communication requirements', *Transportation and Power Grid in Smart Cities*, pp.247–268.

Burrows, M., Abadi, M. and Needham, R. (1989) 'A logic of authentication', *ACM SIGOPS Operating Systems Review*, Vol. 23, No. 5, pp.1–13.

Buttyan, L., Staamann, S. and Wilhelm, U. (1998) 'A simple logic for authentication protocol design', *Proceedings of the 11th IEEE Computer Security Foundations Workshop*, IEEE, USA.

Carman, D., Kruus, P. and Matt, B. (2000) *CONSTRAINTS AND APPROACHES FOR DISTRIBUTED SENSOR NETWORK SECURITY*. Available online at: <https://people.cs.vt.edu/~kafura/cs6204/Readings/SensorNetworks/SensorNetSecurity-NAILabs.pdf> (accessed on 8 June 2021).

Chan, H. and Perrig, A. (2003) 'Security and privacy in sensor networks', *Computer*, Vol. 36, No. 10, pp.103–105.

Chatterjee, S. and Das, A. (2014) 'An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks', *Security and Communication Networks*, Vol. 8, No. 9, pp.1752–1771.

Chatterjee, S. and Roy, S. (2018) 'An efficient dynamic access control scheme for distributed wireless sensor networks', *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 27, No. 1, pp.1–18.

Chatterjee, S., Das, A. and Sing, J. (2015) 'A secure and effective access control scheme for distributed wireless sensor networks', *International Journal of Communication Networks and Distributed Systems*, Vol. 14, No. 1, pp.40–73.

Chuang, Y. and Tseng, Y. (2010) 'An efficient dynamic group key agreement protocol for imbalanced wireless networks', *International Journal of Network Management*, Vol. 20, No. 4, pp.167–180.

Clarke, E., Jha, S. and Marrero, W. (1998) 'Using state space exploration and a natural deduction style message derivation engine to verify security protocols', *Programming Concepts and Methods (PROCOMET'98)*, pp.87–106.

Cremers, C. (2008) 'The Scyther Tool: verification, falsification, and analysis of security protocols', *Computer Aided Verification*, pp.414–418.

Das, A., Chatterjee, S. and Sing, J. (2013) 'A novel efficient access control scheme for large – sclae distributed wireless sensor networks', *International Journal of Foundations of Computer Science*, Vol. 24, No. 5, pp.625–653.

Dolev, D. and Yao, A. (1983) 'On the security of public key protocols', *IEEE Transactions on Information Theory*, Vol. 29, No. 2, pp.198–208.

Eschenauer, L. and Gligor, V. (2002) 'A key-management scheme for distributed sensor networks', *Proceedings of the 9th ACM Conference on Computer and Communications Security*.

Gura, N., Patel, A., Wander, A., Eberle, H. and Shantz, S. (2004) 'Comparing elliptic curve cryptography and RSA on 8-bit CPUs', *Lecture Notes in Computer Science*, pp.119–132.

Hankerson, D., Vanstone, S. and Menezes, A. (2004) *Guide to Elliptic Curve Cryptography*, Springer-Verlag, New York.

Hu, Y.C., Perrig, A. and Johnson, D. (2006) 'Wormhole attacks in wireless networks', *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 2, pp.370–380.

Huang, H. (2009) 'A novel access control protocol for secure sensor networks', *Computer Standards and Interfaces*, Vol. 31, No. 2, pp.272–276.

Huang, H. (2011) 'A new design of access control in wireless sensor networks', *International Journal of Distributed Sensor Networks*, Vol. 7, No. 1, pp.1–8.

Hyun-Sung, K. and Sung-Woon, L. (2009) 'Enhanced novel access control protocol over wireless sensor networks', *IEEE Transactions on Consumer Electronics*, Vol. 55, No. 2, pp.492–498.

Iqbal, U. and Mir, A.H. (2020a) 'Secure and practical access control mechanism for WSN with node privacy', *Journal of King Saud University – Computer and Information Sciences*.

Iqbal, U. and Mir, A.H. (2020b) 'Secure and scalable access control protocol for IoT environment', *Internet of Things*, Vol. 12. Doi: 10.1016/j.iot.2020.100291.

Iqbal, U. and Mir, A.H. (2021c) 'Efficient and dynamic access control mechanism for secure data acquisition in IoT environment', *International Journal of Computing and Digital Systems*, pp.9–28.

Islam, S. and Biswas, G. (2017) 'A pairing-free identity-based two-party authenticated key agreement protocol for secure and efficient communication', *Journal of King Saud University – Computer and Information Sciences*, Vol. 29, No. 1, pp.63–73.

Jamalipour, A. and Zheng, J. (2007) *Wireless Sensor Networks*, Wiley.

Karlof, C., Sastry, N. and Wagner, D. (2004) 'TinySec', *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp.162–175.

Kim, K. and Hong, S. (2013) 'Privacy care architecture in wireless sensor networks', *International Journal of Distributed Sensor Networks*, Vol. 9, No. 5.

Lasassmeh, S. and Conrad, J. (2010) 'Time synchronization in wireless sensor networks: a survey', *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)*, IEEE, USA.

Lee, H., Shin, K. and Lee, D. (2012) 'PACPs: practical access control protocols for wireless sensor networks', *IEEE Transactions on Consumer Electronics*, Vol. 58, No. 2, pp.491–499.

Levis, P. and Gay, D. (2009) *TinyOS Programming*.

Liu, A. and Ning, P. (2008) 'TinyECC: a configurable library for elliptic curve cryptography in wireless sensor networks', *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN'08)*, IEEE, USA.

Malan, D., Welsh, M. and Smith, M. (2008) 'Implementing public-key infrastructure for sensor networks', *ACM Transactions on Sensor Networks*, Vol. 4, No. 4, pp.1–23.

Menezes, B. (2012) *Network Security and Cryptography*, Delmar, Albany, NY.

Mo, J. and Chen, H. (2019) 'A lightweight secure user authentication and key agreement protocol for wireless sensor networks', *Security and Communication Networks*, pp.1–17.

Mote Works (2013) *Getting Started Guide*, PN: 7430-0102-02.

Parno, B., Perrig, A. and Gligor, V. (2005) 'Distributed detection of node replication attacks in sensor networks', *Proceedings of the IEEE Symposium on Security and Privacy*, IEEE, USA.

Perrig, A., Stankovic, J. and Wagner, D. (2004) 'Security in wireless sensor networks', *Communications of the ACM*, Vol. 47, No. 6, pp.53–57.

Perytons Protocol Analyzer (2011) *User Manual*.

Shen, J., Moh, S. and Chung, I. (2010) 'Comment: enhanced novel access control protocol over wireless sensor networks', *IEEE Transactions on Consumer Electronics*, Vol. 56, No. 3, pp.2019–2021.

Shnayder, V., Hempstead, M., Chen, B., Allen, G. and Welsh, M. (2004) 'Simulating the power consumption of large-scale sensor network applications', *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp.188–200.

Wen, M., Lei, J., Li, J., Wang, Y. and Chen, K. (2011) 'Efficient user access control mechanism for wireless multimedia sensor networks', *Journal of Computational Information Systems*, Vol. 7, No. 9, pp.3325–3332.

Zeng, P., Choo, K. and Sun, D. (2010) 'On the security of an enhanced novel access control protocol for wireless sensor networks', *IEEE Transactions on Consumer Electronics*, Vol. 56, No. 2, pp.566–569.

Zhou, Y., Zhang, Y. and Fang, Y. (2007) 'Access control in wireless sensor networks', *Ad Hoc Networks*, Vol. 5, No. 1, pp.3–13.

APPENDIXES

Appendix A: HLPSL model of the proposed access control scheme

```
%%%% ----------Role of a New node played by N_I ----------%%%%
role role_NewNode (NDI:agent, NDJ:agent, G:text,
MUL:function, SND, RCV:channel(dy))
played_by NDI
def=
        local
State:nat, KB1:text, KB2:text, KB:text, KI:text, DVI:text,
RN:text, ndi:text, KJ:text, DVJ:text, ndj:text,
ADD:function, H, INV:function
const Key_KI, NDI_NDJ, NDJ_NDI: protocol_id
  init
State := 0
Transition
    1. State=0 /\ RCV(start) =|> State':=1 /\ KB1':=new()
       /\ KB2':=new() /\ KB':=new() /\ KI' := new() /\
       DVI' := new() /\  ndi':=new()/\ SND(MUL(KI,
       KB1',G).MUL(KI', KB2', H(ndi',
       DVI'),G).ndii'.DVI'.MUL(KI', G)
       .H(MUL(KI', KB1', G).MUL(KI', KB2', H(ndi',
       DVI'), G ).Si'.DVI'.MUL(KI', G)))
       /\   secret(KI', seed_Ki,{NDI})
    2. State=1 /\  RCV(MUL(KJ',KB1',G).MUL(KJ',
       KB2', H(ndj', DVI'),G).ndj'.DVJ'.MUL(KJ', G)
       .H(MUL(KJ' ,KB1', G).MUL(KJ', KB2',
       H(ndj',DVJ'),G ). ndj'.DVJ'.MUL(KJ', G)). {RN'}_
       H (MUL(KI, KJ', KB', ndi, ndj')))
       /\ request (NDI, NDJ, NDJ_NDI, RN') =|>
       State':=2
end role

%%%% ----------Role of a New node played by N_I ----------%%%%
role role_NeighNode (NDI:agent, NDJ:agent, G:text,
MUL:function, SND, RCV:channel(dy))
played_by NDJ
def=
        local
State:nat, KB1:text, KB2:text, KB:text, KI:text, DVI:text,
RN:text, ndi:text, KJ:text, DVJ:text, ndj:text,
ADD:function, H, INV:function
const  Key_KI, NDI_NDJ, NDJ_NDI: protocol_id
init
State := 0
transition
    1.  State=0/\RCV(MUL(KI', KB1', G).MUL(KI',
        KB2', H(ndi', DVI'),G).ndi'.DVI'.MUL(KI',
        G).H(MUL(KI', KB1', G) .MUL(KI', KB2', H(ndi',
        DVI'),G ).ndi'.DVI'.MUL(KI', G))) =|> State':=1
        /\KJ' := new() /\ ndj':= new() /\ KB1':=new()
        /\KB2':=new() /\ KB':=new() /\ RN':= new()
        /\SND( MUL(KJ', KB1', G).MUL(KJ', KB2',
        H(ndj',DVI'),G ).ndj'.DVJ'.MUL(KJ', G).
        H(MUL(KJ', KB1', G). MUL(KJ', KB2',H(ndj',
        DVJ'),G ).ndj'.DVJ'.MUL(KJ', G)).{RN'}_H(MUL
        (KI, KJ', KB', ndi, ndj')))
        /\ witness(NDJ, NDI, NDJ_NDI, RN')/\secret(KJ',
Key_KI, {NDJ})
end role

%%%% ----------
Role Session----------
%%%%
role session (NDI:agent, NDJ:agent, G:text, MUL:function)
def=
local
SND2, RCV2, SND1, RCV1:channel(dy)
Composition
        role_NeighNode (NDI, NDJ, G, MUL, SND2,
RCV2) /\ role_NewNode (NDI, NDJ, G, MUL, SND1,
RCV1)
end role

%%%% ----------
Role Environment ----------
%%%%
role environment()
def=
        const
                bob:agent, mul:function, alice:agent,
g:text
        intruder_knowledge = {alice, bob, g}
        composition
                 session(alice, bob, g, mul)
end role

%%%% ----------
Goal Section----------
%%%%
goal
secrecy_of KI
secrecy_of KJ
authentication_on NDI_NDJ
authentication_on NDI_NDJ
end goal
environment()
```

Appendix B: SPDL model of the proposed access control scheme

```
hashfunction MUL;
hashfunction H;
hashfunction ADD;
protocol proposed-protocol (I, R)
{
```

```
####----------------------------------------------------Role I
playing as New Node NI ----------------------------------------
--####
 role I
        {
         secret Kb1, Kb2, Kb;
         fresh Ki, Ni, DVi, G: Nonce;
         var  Kj, Nj, DVj, Rn, G: Nonce;

         send_1 (I, R, MUL(Ki, Kb1, G),
        MUL(Ki, Kb2, H(Ni, DVi),G), Ni,
        DVi, MUL(Ki, G), H(MUL(Ki, Kb1, G),
        MUL(Ki, Kb2, H(Ni, DVi),G), Ni, DVi,
        MUL(Ki, G)));
        recv_2(R, I, MUL(Kj, Kb1, G) , MUL
        (Kj, Kb2, H (Nj, DVj),G), Nj, DVj,
        MUL(Kj, G), H(MUL(Kj, Kb1, G) ,
        MUL(Kj, Kb2, H(Nj, DVj), G), Nj, DVj,
        MUL(Kj, G)), {Rn}H (MUL(Ki, Kj, Kb,
        DVi, DVj, G)), H(Rn));
        claim_i1(I, Secret, Ki);
        claim_i2(I, Alive);
        claim_i3(I, Weakagree);
        claim_i4(I, SKR, H(MUL(Ki, Kj,
        Kb, DVi, DVj, G)));
        claim_i5(I, Niagree);
        claim_i6(I, Nisynch);

}
```

```
####----------------------------------------------------Role R
playing as New Node NJ ----------------------------------------
-####
 role R
        {
        secret Kb1, Kb2, Kb;
        fresh Kj, Nj, DVj, Rn, G: Nonce;
        var Ki, Ni, DVi, G: Nonce;

        recv_1(I, R, MUL(Ki, Kb1, G), MUL(Ki, Kb2,
        H(Ni, DVi), G), Ni, DVi, MUL(Ki, G),
        H(MUL(Ki, Kb1, G), MUL(Ki, Kb2,
        H(Ni, DVi), G), Ni, DVi, MUL(Ki, G)));
        send_2(R, I, MUL(Kj, Kb1, G), MUL(Kj, Kb2,
        H(Nj, DVj), G), Nj, DVj, MUL(Kj, G),
        H(MUL(Kj, Kb1, G) , MUL(Kj, Kb2,
        H(Nj, DVj), G), Nj, DVj, MUL(Kj,
        G)),{Rn}H(MUL(Ki, Kj, Kb, DVi, DVj, G)),
        H(Rn));
        claim_r1(R, Secret, Kj);
         claim_r2(R, Alive);
         claim_r3(R, Weakagree);
        claim_r4(R, SKR, H(MUL(Ki, Kj, Kb,
        DVi, DVj, G)));
        claim_r5(R, Niagree);
        claim_r6(R, Nisynch);

        }
}
```