International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 10, No.1 (Jan-2021)
http://dx.doi.org/10.12785/ijcds/100102

Efficient and Dynamic Access Control Mechanism for Secure Data Acquisition in IoT Environment

Ummer Iqbal [1]and Ajaz Hussain Mir[1]

[1]*National Institute of Technology, Srinagar, India*

Received 5 Feb. 2020, Revised 12 Apr. 2020, Accepted 25 Jul. 2020, Published 1 Jan. 2021

Abstract: Wireless Sensor Network **(**WSN) is an essential constituent of IoT based smart city applications.WSN is one of the prominent sensing technologies for data acquisition in smart surveillance and monitoring applications. However, the deployment of a new node in the WSN is a critical security concern. The new node deployment within WSN is inevitable due to the outage of power or nodes getting compromised by adversary attacks. The newly deployed node can be a malicious node that may disrupt and compromise the data acquisition process of IoT based smart applications. Thus for effective and secure new node deployment in WSN, an access control mechanism needs to be enforced. Many schemes have been suggested for access control within WSN, but the practical consideration has been ignored. In this paper, a secure and practical access control mechanism based on Elliptical Curve Cryptography (ECC) has been presented for secure data acquisition using WSN. The proposed access control scheme highlights and address the practical implementation issues, which include scalability and no interdependence on clock synchronization between the nodes in a WSN. A detailed comparative analysis with the existing scheme suggests a better tradeoff in terms of security and functional requirements as compared to the existing relevant schemes. The security strength of the proposed scheme has been formally validated using Scyther. Scyther analysis depicts that the proposed scheme is resistant against various active and passive attacks. The proposed scheme has also been simulated on TinyOS using the TOSSIM simulator to carry out a detailed energy analysis. Based on the simulation analysis, a new node requires only 178 mJ to join the WSN based data acquisition tier. The formal validation and simulation results suggest that the proposed scheme can be used for secure data acquisition using WSN for IoT based smart applications.

Keywords: IOT,WSN,Acces Control ,ECC,Scyther,TinyOS,TOSSIM

1. INTRODUCTION

The Wireless Sensor Network has gained huge momentum over the period owing to its importance in IoT based smart city applications [1] The various smart applications based on IoT include: Smart Traffic, Smart homes, Smart Offices, Smart Grids, Smart health, Environmental Monitoring, and Surveillance System[2] [3][4]. According to a report in 2016 [5], the market projections of WSN were valued to be $29.06 billion and is predicted to reach $93.86 billion by 2023. The surveillance and monitoring applications based on IoT/WSN are primarily divided into 3 Tiers: Data Acquisition Tier, Network Tier, and Client Tier as depicted in Figure 1[6][7]. The data acquisition tier, also known as the perception layer, is primarily involved in perceiving and sensing various physical parameters. Various sensing technologies that can be employed in the data acquisition tier include WSN, GPS, NFC, etc. A WSN based data acquisition tier primarily comprises of a network of Data Acquisition Nodes (DAN), also called as motes which are randomly distributed in the region of interest. A typical DAN deployed within the data acquisition tier is characterized by 4 KB of RAM and 128 KB ROM and equipped with 2 AA batteries [8]. DAN is interfaced with various types of sensors depending upon the type of application and the parameters to be monitored.

DAN's communicate with a gateway in a data acquisition tier through a multi-hop network. The DAN's deployed in the data acquisition tier are classified either as a Full Function Device (FFD) or a Reduced Function

E-mail address: ummer.iqbal.khan@gmail.com, ahmir@nitsri.net

Device (RFD). A RFD primarily senses the required parameters of interest. These sensed parameters are forwarded to the WSN gateway by a multi-hop communication performed by the Full Function Devices. The data acquired by a WSN gateway is further relayed to the Network Tier. A Network Tier is responsible for processing and calibration of raw data received from the data acquisition tier. Besides that, it provides communication support for transmitting the data to the client tier through various wired and wireless network technologies, which include LAN, WiFI,4G/3G, etc. As smart applications are associated with a massive amount of data, the network tier also provides the functionalities to store and process the data. The middleware support is provided by various types of servers, which include mobile, web, and real-time communication server. Besides that, for high volume and archival data storage and management, cloud computing technology can also be employed at this layer. The client tier is the front end of the IoT Based applications. It provides a platform for actuations based on the received sensed data. Besides that, the Client Tier involves the visualization and analysis of the data by the intended user. Real-time online or archival data can be visualized on various client terminals, which include PDA, laptop, PC, etc.

The smart IoT applications based on 3-tier architecture, as depicted in Figure 1, needs an integrated security approach. However, security within the WSN based data acquisition tier has been a significant area of research. The WSN based data acquisition tier involves resource constraint devices, thus possessing serve security challenges. As with other resource-constrained sensing technologies, traditional security mechanisms cannot be directly applied to WSN owing to its constraints in terms of energy and memory [9][10]. As a result, securing WSN has become an active area of research.

In terms of various security requirements, Access Control is predominantly one of the essential and significant security mechanism required in WSN [11][12]. An access control mechanism regulates the deployment of a new node in the data acquisition tier. The need for a new DAN deployment arises as the nodes are lost due to adversary attacks or drainage of battery. Deployment of a malicious node in the network can disrupt the complete network as it may lead to significant attacks like False report Injection, Node capture, Sybil Attack, Worm Hole Attack Man in the Middle Attack and Message Replay attack [13]. An access control mechanism primarily encompasses authentication and key establishment mechanism to determine the legitimacy of the new DAN to join the network [13]. The authentication of a new node by its neighbors ensures that only a legitimate node can join a network. After the mutual authentication of the newly deployed DAN with its neighbors, a pair-wise key is also established with them.

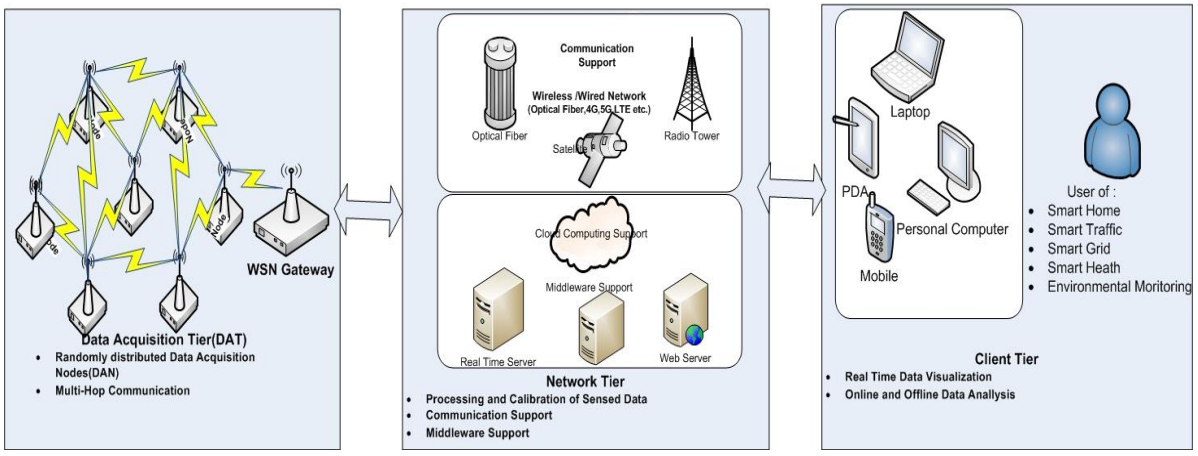


Figure 1. 3-Tier Architecture of Surveillance and monitoring applications based on WSN

The pair-wise key helps a new DAN to have secure communication with its neighbors. Any access control scheme targeting WSN must oblige to its constraints in terms of computation and memory. Many schemes have been suggested for access control in WSN with low overheads in terms of computation, memory, and communication. However, practical considerations and requirements of an access control scheme for WSN have mostly been ignored.

A. Major Requirements for Practical Access Control Mechanism

For an access control scheme to be considered for practical implementation, the major requirements are discussed below.

a) R1: Low overhead in terms of Computation, Communication, and Memory:

In order to serve in limited resources, an access control mechanism must be functionally optimal in terms of computational and communication overhead's[14]. An access control mechanism must typically involve less

number of bits transmitted/received as energy consumed in communication is three times greater than that required for computation[15].

b) R2: Must be scalable:

Scalability is one of the most important functional requirements for the practical implementation of an access control scheme. The design of an access control scheme must not require the involvement of a base station for new node addition. The involvement of the base station is required for new node addition when a base station requires to broadcast the new node parameters to the complete network. The involvement of the Base Station significantly increases the communication overhead, thus making the scheme infeasible for practical implementation[13]. Moreover, a scalable access control scheme does not require the re-deployment or bootstrapping of the other nodes in the network for new node deployment.

c) R3: Must not be dependent on Time Synchronization

As an access control scheme involves a request for new node addition to a network, it becomes prudent to distinguish whether the new node deployment request is fresh or obsolete. Typically, many schemes use timestamps to evaluate the freshness of the deployment request, which requires clock synchronization between the node in the network. Achieving clock synchronization within WSN is itself a complex research issue owing to its resource constraint nature[16]. Furthermore, clock synchronization between the nodes in WSN requires a high computational and communication overhead. Thus a practical access control scheme must not ideally depend on clock synchronization between the nodes in the network as it increases its complexity and severely limits its efficacy.

d) R4: Must be secure and formally validated against various active and passive attacks:

A practical and secure access-control must be resilient to various active and passive attacks. The significant attacks to which a practical access control mechanism must be resilient to include: Sybil Attack, Worm Hole Attack Node replication Attack, Man in the Middle Attack and False report Injection Attack[13]. Besides that, for considering an access control scheme for practical implementation, it must be formally validated against various active and passive attacks in general. Automated formal validation tools can be used to determine the security strength of proposed access control and highlight its consideration for practical implementation.

B. Motivation

An access control mechanism for deployment of a new Data Acquisition Node (DAN) in the WSN is a security primitive of significant importance. Schemes have been suggested in the literature for new node deployment with a strong emphasis on low overheads in terms of computation and memory. However, for the practical implementation of an access control scheme, the requirements of scalability and no interdependence on clock synchronization has not been paid much attention. It is pertinent to mention that for the practical realization of the access control scheme, it must be feasible for large WSN, thus providing a high degree of scalability.

Clock synchronization within WSN is also an emerging research issue owning to its resource constraint nature. Thus, the dependence of an access control scheme on clock synchronization between the nodes in the network restricts its practical applicability. The primary motivation behind the proposed work is to propose a practical access control scheme for new node addition with a strong emphasis on scalability and no interdependence on clock synchronization between the nodes in the network.

C. Contributions

In this paper, a practical and secure access control mechanism for new DAN deployed within the WSN based data acquisition tier has been proposed. The major highlights of the proposed scheme are given below:

1. The proposed scheme provides an efficient and secure access control mechanism for new node deployment in WSN based Data Acquisition Tier with better trade-off as compared to the existing related schemes.

2. The proposed scheme suffices the major requirements for the practical implementation of access control, which includes scalability and no interdependence on time synchronization.

3. The security strength of the proposed scheme is formally verified and validated using Scyther[17]. The Scyther results determine that the scheme is SAFE.

4. The proposed scheme has been simulated on TinyOS[18] platform using the TOSSIM simulator. A detailed energy analysis of the proposed scheme has also been carried out using Power TOSSIM[19]. The simulation study determines that the scheme requires only 178 mJ for new node addition.

The rest of the paper is organized as follows. Section 2 presents the preliminary background of elliptical curve cryptography. Section 3 reviews and presents the drawback of existing access control schemes. Section 4 presents the proposed access control based on elliptical curve cryptography for dynamic new DAN addition. Section 5 provides the formal validation of the scheme against various active and passive attacks using the

Scyther tool. Section 6 presents a detailed comparison of the proposed scheme with the relevant existing schemes. Section 7 provides the implementation details of the proposed scheme on TinyOS and highlights the energy requirements of the proposed protocol using PowerTOSSIM.

2. ELLIPTICAL CURVE CRYPTOGRAPHY

Elliptical curve cryptography (ECC) was proposed by Koblitz [20] and Miller [21]. ECC is a highly efficient cryptosystem with low overheads as compared to traditional systems like RSA. Elliptical curve cryptography, with the key size of 160-bit key size, provides the same level of security as that of the key size of 1024 bits in RSA, which makes it suitable for resource constraint devices[22].

An Elliptical Curve E(a,b) over a finite prime field F_p is defined as (1) :

$$E(a,b) : y^2 = x^3 + ax + b \quad (1)$$

The curve E(a,b) is characterized by domain parameters D={ a,b,G(x,y),n,h) . [a,b] are the coefficients of the curve: E(a,b) such [a,b] € F_p. G(x,y) is the generator point of the curve, such that [x,y] € F_p. n is the order of the curve, and h is the cofactor. For E(a,b) to be smooth and have no degeneration, the determinant (▲) must be other than zero, as depicted in (2).

$$▲ = (4a^3 + 27b^2 \neq 0) \quad (2)$$

Some of the important definitions related to an elliptical curve operation are listed below[22][23]:

1. **Point Addition** : For any two points [A(x,y), B(x,y)] € E(a,b) , the point addition (+) is performed as (3) :

   $$C(x,y) = A(x,y) + B(x,y) \quad (3)$$

   Where C(x,y) € E(a,b) and is a reflection of the point where line joining A(x,y) and B(x,y) intersects the curve E(a,b)

2. **Scalar Multiplication:** For any scalar X € Z_q and P(x,y) € E(a,b), the scalar multiplication (.) is defined as(4)

   $$X * P(x,y) = \underbrace{P(x,y) + P(x,y) + P(x,y) + \ldots + P(x,y)}_{X \text{ times}} \quad (4)$$

3. **Elliptical Discrete Logarithmic Property (ECDLP):** Given two points P(x,y) and Q(x,y) € E(a,b) as given in (5)

   $$Q(x,y) = n.P(x,y) \quad (5)$$

   Where n is a scalar, ECDLP states that it is computationally infeasible to find n. ECDLP has an exponential running time complexity.

3. LITERATURE REVIEW

Zhou et al. [24] proposed an access control scheme based on elliptical curve cryptography. In this scheme, preloaded ECC certificates were used to achieve authentication and key establishment between the new nodes and their neighbors. The scheme obliged to the important security requirements of an access control mechanism. However, it needs time synchronization between the nodes in the network for its implementation. Moreover, the Zhou et al. scheme has high computational and communication costs. In 2009, Huang [25] suggested the Novel Access Control Protocol (NACP) scheme based on hash chains and elliptical curve cryptography. Huang's scheme is susceptible to various security attacks, which include Node Replication Attack, Man in the Middle Attack, False Report Injection, and Replay attack. The scheme also requires the intervention of the base station for each new node addition. In Huang's scheme, the base station is required to broadcast the public hash commitments of the newly added node within the network. However, the scheme did not require time synchronization between the nodes in the network for its implementation. The scheme has not been formally validated against various active and passive attacks.

Kim and Lee suggested Enhanced Novel Access Control Protocol (ENACP)[26] overcome the limitations of NACP. ENACP addressed the issue of hash chain renewability of NACP. However, the scheme is not scalable and also has not been formally validated against various active and passive attacks: Zeng et al. [27] and Sheng et al.[28] further evaluated ENACP to point out significant security limitations in the scheme. In 2011, Huang[29] proposed an access control scheme based on ECC and one-way hash chains. The scheme is scalable but requires clock synchronization between the nodes in the network for its implementation. The scheme has not been formally validated against various active and passive attacks. In 2012, Lee et al. suggested a Practical Access Control Protocol for wireless sensor networks (PACP)[30]. Lee et al. pointed out that hash chain based access control schemes cannot be used for practical implementation as they require maintaining the state of the network. However, PACP requires a significant

memory overhead as it involves the storage of pre-deployed keys in the nodes. In 2013 Das et al. [31] proposed an access control scheme based on preloaded ECC signatures. The scheme has been formally validated against various active and passive attacks. However, it requires time synchronization for its implementation and has significantly high computational overhead.

In 2014, Chatterjee et al. [32] highlighted the deficiencies of Huang (2011) and suggested an access control scheme to overcome them. The scheme has been validated against various active and passive attacks. The scheme has high computational overhead and requires time synchronization between the nodes in the network for implementation. In 2015, Chatterjee et al. [33] suggested another scheme based on one-way hash functions. The suggested scheme has very low overheads in terms of computational and communication overheads. The scheme has also been verified using AVISPA. However, the scheme is not scalable as it requires the intervention of a base station for new node addition. Another drawback of the scheme is that it requires clock synchronization between the nodes for its implementation. In 2018, Chatterjee et al[13] . suggested an efficient and secure access control scheme for new node addition in WSN. The scheme is scalable and has been formally validated against various active and passive attacks. The suggested scheme is scalable; however, it requires time synchronization between the nodes in the network for its implementation [9]. The limitation of the existing schemes is tabulated in Table 1. The proposed access control scheme is designed to add a new node in a network while addressing the limitations highlighted in the existing schemes.

Table 1. Limitations of Existing Schemes

Scheme	Limitations in terms of Practical Applicalaitby
Zhou et al. (2007)[24]	The scheme has a very high resource overhead in terms of communication, computation and memory. The scheme is scalable but requires time synchronization for its implementation. The scheme has not been formally validated against active and passive attacks
Huang (2009)[25]	The scheme has a very high resource overhead in terms of communication, computation and memory. The scheme is scalable but requires time synchronization for its implementation. The scheme has not been formally validated against active and passive attacks
Kim and Lee (2009)[26]	The scheme is not scalable but is independent of the time synchronization issue. The scheme is susceptible to various security attacks, which include false report injection, node capture, Man in the Middle Attack, and message replay attack. The scheme is not formally validated against various active and passive attacks
Huang (2011)[29]	The scheme is not scalable but is independent of the time synchronization issue. The scheme is susceptible to various security attacks, which include false report injection, node capture, Man in the Middle Attack, and message replay attack. The scheme is not formally validated against various active and passive attacks
Das et al. (2013)[31]	The scheme has high computational overhead and requires time synchronization between the nodes in the network for its implementation
Chatterjee et al. (2015) [33]	The scheme is not scalable and requires time synchronization between the nodes in the network for its implementation
Chatterjee et al. (2018) [13]	The scheme requires time synchronization between the nodes in the network for its implementation

4. PROPOSED ACCESS CONTROL SCHEME

In this section, a dynamic access control scheme for new DAN addition in the WSN based Data Acquisition Tier is presented. The scheme is based on elliptical curve cryptography and comprises 3 phases: 1.Set-up 2. Initialization 3.DAN authentication and key establishment The notations used in the proposed scheme are listed in Table 2.

Table 2. Notations

Symbol	Description
E (a,b)	Elliptical Curve
GWN	Gateway Node
DAN	Data Acquisition Node
DAN_I	Identity of Data Acquisition Node I
DAN_J	Identity of Data Acquisition Node J
K_I	Random Secret of DAN_i
Sp	Private Key of GWN
$G_{pub}(x,y)$	Public Key of GWN
G(x,y)	Generator Point of Elliptical Curve E(a,b)
SD_i	Set Deployment Identifier of DAN_i
SD_i^j	Last Seen Set Deployment Identifier of DAN_i maintained by DAN_J
K_{ij}	Session Key between N_i and N_j
R_{ID}	Deployment Request Type
.	Scalar Multiplication
+	Point Addition

A. *Network and Attack Model*

The network model of the data acquisition tier is based on flat WSN topology, as depicted in Figure 1. A deployment request from a DAN can be either a fresh deployment or a re-deployment request. A fresh deployment request from DAN is generated when it is being deployed for the first time in a particular region of interest. A re-deployment request is sent by a DAN in case it needs to get re-deployment due to the outage of

power or being compromised due to adversary attacks. R_{ID} is used to distinguish between the two types of requests. If R_{ID}=0, it is a fresh deployment request, and R_{ID} =1 is for the re-deployment. DAN's in the data acquisition tier is deployed in sets. Each set of nodes deployed within the region of interest is identified by the set deployment identifier. Set deployment identifier acts as a marker to distinguish between the old node and the new node. For all the DAN's deployed in the first set, the set deployment identifier is set to 1. Each node maintains a last seen set deployment identifier of its neighbors' in the communication range. The last seen set deployment identifier is used to distinguish the old and a new node in case of redeployment. For a DAN_I , it's set deployment identifier is SD_I, and the last seen set deployment identifier in DAN_J is identified by DAN_I^J as indicated in Table 2.

The attack model considered in the scheme is the Dolev and Yoa model[26]. In Dolev and Yoa model communication model is considered to be insecure. The adversary can intercept, masquerade, and modify the data[27]. The nodes in the Data Acquisition Tier are susceptible to node capture attacks as they are not temper resistant. However, GWN is highly secure and has high computational resources.

B. Phase 1-Set-up phase

During the Setup phase, GWN performs the following steps:

1. GWN chooses the parameters of an Elliptical Curve E (a, b): $y^2= x^3+ax+b$. The parameters a, b, must be chosen in such a way that $4ax^3+27b^2 \neq 0$.

2. GWN computes its public key $G_{pub}(x,y)$, where $G_{pub}(x,y)= Sp. G(x,y)$

3. The Parameters of E(a,b) and $G_{pub}(x,y)$ are made public.

C. Phase2- Initialization phase

The initialization phase is carried out before the deployment of DAN in the Data Acquisition Tier. The following steps are performed by the GWN.

1. GWN chooses a random secret for each K_I for each DAN_I and computes C_I as (6)

$$C_I = K_I . G(x,y) \quad (6)$$

2. GWN divides K_I into random unequal parts as K_I^1 and K_I^2 such that $K_I = K_I^1 + K_I^2$

3. GWN divides Sp into random unequal parts as Sp^1 and Sp^2 such that $Sp = Sp^1 + Sp^2$

4. GWN computes the split signature pair (S_1^I, S_1^2) of the DAN_I to be deployed in the region of interest as (7) and (8):

$$S_1^I = (Sp^1 + K_I^1).H(DAN_I \| SD_I). G(x,y) \quad (7)$$
$$S_2^I = (Sp^2 + K_I^2).H(DAN_I \| SD_I). G(x,y) \quad (8)$$

5. Each DAN_I in the network is preloaded with the following information:
   a) Elliptical Curve E(a,b)
   b) Hash Function H()
   c) Split Signature Pair (S_1^I, S_1^2) and $C_I(x,y)$
   d) The public key of the GWN: $G_{pub}(x,y)$
   e) Data Acquisition Node Identity: DAN_I
   f) Set Deployment Identifier: SD_I

The Sequence of steps are summarized in Figure 2.

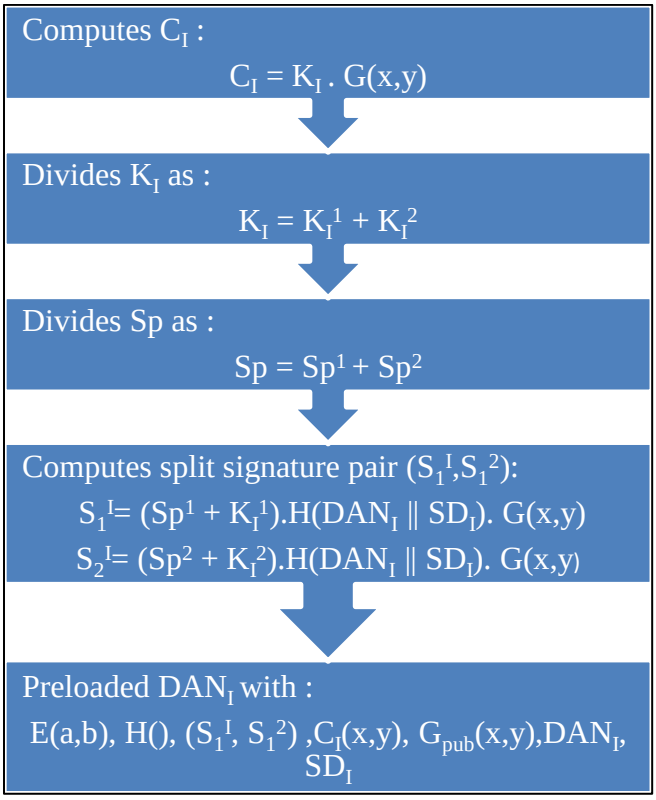


Figure 2. Steps in the initialization phase

D. Phase3-DAN authentication and key establishment

This phase comprises of authentication and key establishment of a newly deployed DAN with its neighbors in the communication range. The

authentication mechanism is employed to determine whether a newly deployed DAN is a legitimate node and can join the data acquisition tier. After the legitimacy of the newly deployed DAN is authenticated, it establishes a pairwise symmetric key with all its neighbors for secure communication with them. Let DAN_I be the new node that wants to join the Data Acquisition Tier. After the DAN_I is deployed in the region of interest, it broadcast the deployment request to all its neighbors in the communication range.

$$DAN_I \rightarrow * : (C_I(x,y)\ S_1^I(x,y),\ S_2^I(x,y)\ ,DAN_I,\ SD_I,\ R_{ID})||\ H[(C_I(x,y)\ S_1^I(x,y),\ S_2^I(x,y),DAN_I,\ SD_I,\ R_{ID}]$$

The Broadcast is received by all neighboring nodes of the DAN_I. Let DAN_J be the neighboring DAN who receives the broadcast from DAN_I. DAN_J evaluates the deployment request-id R_{ID} to determine whether the request from DAN_I is the fresh deployment or Re-Deployment. Based on the value of the R_{ID} following cases may be evaluated by DAN_J:

1. *Request is a Fresh Deployment*
2. *Request is a Re-Deployment*

1. *Request is the Fresh deployment*

In this case, DAN_I is being deployed for the first time in the data acquisition tier, as indicated by $R_{ID}= 0$. DAN_J receives the following fresh deployment broadcast from the DAN_I:

$$(C_I(x,y)\ S_1^I(x,y),\ S_2^I(x,y)\ ,DAN_I,\ SD_I,\ R_{ID}=0)||H[(C_I(x,y)\ S_1^I(x,y),S_2^I(x,y),DAN_I,\ SD_I,\ R_{ID}=0]$$

DAN_J evaluates the fresh deployment request and the request is accepted or rejected based on the following cases:

Case 1: if $SD_I < SD_J$ is true then DAN_I is considered as the old node and the request is rejected. The rejection is because the set deployment identifier of DAN_I is less than that of DAN_J indicating that DAN_I request for fresh deployment is obsolete and as such no DAN authentication and key exchange handshake is initiated

Case 2: if $SD_I > SD_J$ is true then DAN_I is considered as the new node and the request is accepted. The acceptance is because the set deployment identifier of DAN_I is greater than that of DAN_J, indicating that DAN_I is deployed after DAN_J. As the request is accepted, DAN authentication and key exchange handshake is initiated.

Case 3: if $SD_I == SD_J$ is true then DAN_I is considered as the new node and the request is accepted. The acceptance is because of the fresh deployment and, as such, DAN_J can identify that it has not maintained any last seen set deployment identifier SD_I^J for DAN_I. Thus no DAN authentication and key exchange handshake have previously taken place between DAN_I and DAN_J, which implies that DAN_I and DAN_J are deployed in the same set. As the request is accepted, DAN authentication and key exchange handshake is initiated.

2. *Request is a Re-Deployment*

The Re-Deployment of DAN_I in a particular region of interest within the data acquisition tier may be due to batteries getting exhausted or the node getting compromised due to an adversary attack. DAN_J receives the following Re-Deployment broadcast from the DAN_I:

$$(C_I(x,y)\ S_1^I(x,y),\ S_2^I(x,y)\ ,DAN_I,\ SD_I,\ R_{ID}=1)\ ||H[(C_I(x,y)\ S_1^I(x,y),S_2^I(x,y),DAN_I,\ SD_I,\ R_{ID}=1]$$

DAN_J evaluates the Re-Deployment request, keeping in consideration that the request needs to be evaluated by comparing SD_I^J and SD_I. The request is accepted or rejected based on the comparison of SD_I and SD_I^J as given below:

Case 1: if $SD_I < SD_I^J$ is true, then the re-deployment request is considered as an old request and rejected. In this case, the set deployment identifier (SD_I) in the current request is older than that of the last seen set deployment identifier (SD_I^J) maintained by DAN_J, which implies that the request is obsolete. As the request is rejected, no DAN authentication and key exchange handshake is initiated

Case 2: if $SD_I > SD_I^J$ is true, then the DAN_I re-deployment request is considered to be a new request. The request is accepted as the comparison indicates that the current request is a recent deployment as compared to the deployment identifier of DAN_I already maintained by DAN_J. As the request is accepted, DAN authentication and key exchange handshake are initiated.

Case 3: if $SD_I == SD_I^J$ is true, then the DAN_I request is considered to be an old request. The set deployment identifier of the current request is the same as that maintained by DAN_J, indicating that the request has been previously handled and accepted. As the request is rejected, no DAN authentication and key exchange handshake is initiated

E. DAN authentication and key exchange handshake

After the deployment request of DAN_I is accepted by DAN_J following steps are undertaken in DAN authentication and key exchange handshake:

1. After accepting the deployment request from DAN_I, DAN_J verifies the integrity of the received request. DAN_J computes the hash of the received request: $(C_I(x,y)\ S_1^I(x,y), S_2^I(x,y), DAN_I, SD_I, R_{ID})$ as $H''[(C_I(x,y)\ S_1^I(x,y), S_2^I(x,y), DAN_I, SD_I, R_{ID}]$. The received hash and the hash computed by DAN_I are compared:

$$S_I(x,y)=(Sp^1 + Sp^2 + K_I^1 + K_I^2).H(DAN_I \| SD_I).\ G(x,y) \quad (8)$$

$$S_I(x,y) =(Sp + K_I).H(DAN_I \| SD_I).\ G(x,y) \quad (9)$$

$$S_I(x,y) =(G_{pub}(x,y) + C_I(x,y))\ H(DAN_I \| SD_I) \quad (10)$$

3. DAN_J computes $X= [H\ (DAN_I \| SD_I)]^{-1}$ and performs its scalar multiplication with $S_I(x,y)$ as (11) and (12).

$$S_I(x,y)=(G_{pub}(x,y)+C_I(x,y)).H(DAN_I \| SD_I).X \quad (11)$$

$$S_I(x,y) = (G_{pub}(x,y) + C_I(x,y)) \quad (12)$$

DAN_J further computes $C_I''(x,y)$ by performing a point addition of $S_I(x,y)$ with $(-G_{pub}(x,y))$ as (13) and (14).

$$C_I''(x,y) = S_I(x,y) + (-G_{pub}(x,y)) \quad (13)$$

$$C_I''(x,y) = (G_{pub}(x,y) + C_I(x,y)) + (-G_{pub}(x,y)) \quad (14)$$

In case $C_I''(x,y) \neq C_I(x,y)$, DAN_I authentication has failed. No further processing is done by DAN_J, and DAN addition handshake is aborted. If $(C_I''(x,y) = C_I(x,y))$, then DAN_I is authenticated as a legitimate node. DAN_J further calculates its symmetric key with DAN_I as (15).

$$K_{ij} = H[C_I(x,y) * K_J] \quad (15)$$

DAN_J also sends $(C_J(x,y)\ S_1^J(x,y),\ S_2^J(x,y)\ , DAN_J, SD_J, R_{ID}) \| E_{Kij}\ [H[(C_J(x,y)\ S_1^J(x,y),\ S_2^J(x,y), DAN_J, SD_J, R_{ID})]]$ to DAN_I

4. On receiving $(C_J(x,y)\ S_1^J(x,y),\ S_2^J(x,y), DAN_J, SD_J, R_{ID}) \| E_{Kij}\ [H[(C_J(x,y)\ S_1^J(x,y),\ S_2^J(x,y), DAN_J, SD_J, R_{ID})]]$ from DAN_J, DAN_I computes $S_J(x,y) = S_1^J(x,y) + S_2^J(x,y)$ as (16) (17) and(18):

$$S_J(x,y)=(Sp^1 + Sp^2 + K_J^1 + K_J^2).H(DAN_J \| SD_J).G(x,y) \quad (16)$$

$$S_J(x,y) =(Sp + K_J).H(DAN_J \| SD_J).\ G(x,y) \quad (17)$$

$$H[(C_I(x,y)\ S_1^I(x,y), S_2^I(x,y), DAN_I, SD_I, R_{ID}] \equiv H''[(C_I(x,y)\ S_1^I(x,y), S_2^I(x,y), DAN_I, SD_I, R_{ID}]$$

If the integrity check evaluates to be correct, then the request is accepted, and step 2 is performed. In case the integrity check evaluates to false, no further processing is done, and the request is rejected.

2. DAN_J computes $S_I(x,y) = S_1^I(x,y) + S_2^I(x,y)$ as (8),(9) and (10):

$$S_J(x,y) =(G_{pub}(x,y) + C_J(x,y))\ H(DAN_J \| SD_J \quad (18)$$

5. DAN_I computes $Y=[\ H(DAN_J \| SD_J)]^{-1}$ and performs its scalar multiplication with $S_J(x,y)$ as (19) and(20).

$$S_J(x,y) =(G_{pub}(x,y) + C_J(x,y))\ H(DAN_J \| SD_J).Y \quad (19)$$

$$S_J(x,y) =(G_{pub}(x,y) + C_J(x,y)) \quad (20)$$

DAN_I further computes $C_J''(x,y)$ by performing a point addition of $S_J(x,y)$ with $(-G_{pub}(x,y))$ as (21).

$$C_I''(x,y) = S_J(x,y) + (-G_{pub}(x,y)) \quad (21)$$

In case $C_J''(x,y) \neq C_J(x,y)$, DAN_J authentication has failed. No further processing is done by DAN_I and DAN addition handshake is aborted. If $(C_J''(x,y) == C_J(x,y))$, then DAN_I further calculates its symmetric key with DAN_J as (22).

$$K_{ij} = H[C_J(x,y) * K_I] \quad (22)$$

6. DAN_I decrypts the received $E_{Kij}\ [H[(C_J(x,y)\ S_1^J(x,y), S_2^J(x,y), DAN_J, SD_J, R_{ID})]]$ using K_{ji} as (23) :

$$H_E=D_{Kji}[E_{Kij}\ [H[(C_J(x,y)\ S_1^J(x,y), S_2^J(x,y), DAN_J, SD_J, R_{ID})]]] \quad (23)$$

DAN_I computes the hash of the received request $(C_J(x,y)\ S_1^J(x,y),\ S_2^J(x,y),\ DAN_J,\ SD_J,\ R_{ID})$ as $H''[(C_J(x,y)\ S_1^J(x,y),\ S_2^J(x,y), DAN_J, SD_J, R_{ID})]$. The H_E and the hash computed by DAN_I are compared as (24).

$$H_E \equiv H''[(C_J(x,y)\ S_1^J(x,y),\ S_2^J(x,y), DAN_J, SD_J, R_{ID})] \quad (24)$$

If the check evaluates to be correct then DAN_J is authenticated by DAN_I. In case the check evaluates to false, no further processing is done, and the process is aborted. The flow chart of the complete access control scheme is shown in Figure 3.

5. FORMAL SECURITY VALIDATION AND VERIFICATION USING SCYTHER

Scyther is an automated security protocol validation tool developed by Cremers[17]. A significant highlight of scyther is that it provides unbounded verification of security protocols. Security protocols in scyther are verified by considering the communication channel under the Dolev and Yoa threat model[34][35]. In Scyther, security protocols are modeled in Scyther Protocol Description Language (SPDL). In SPDL, communicating parties are modeled as roles and the communication pattern is specified within these roles. The communication between the specified roles is implemented using send and receive operations. To implement cryptographic functions, scyther has a support of various cryptographic operations, which include hash functions, encryption and decryption functionality, etc. In order to verify the security strength of a protocol, various types of claims are declared within the roles defining the protocol. Claims in Scyther have given below:

1. **Secret:** declares a value that must remain secret during the communication

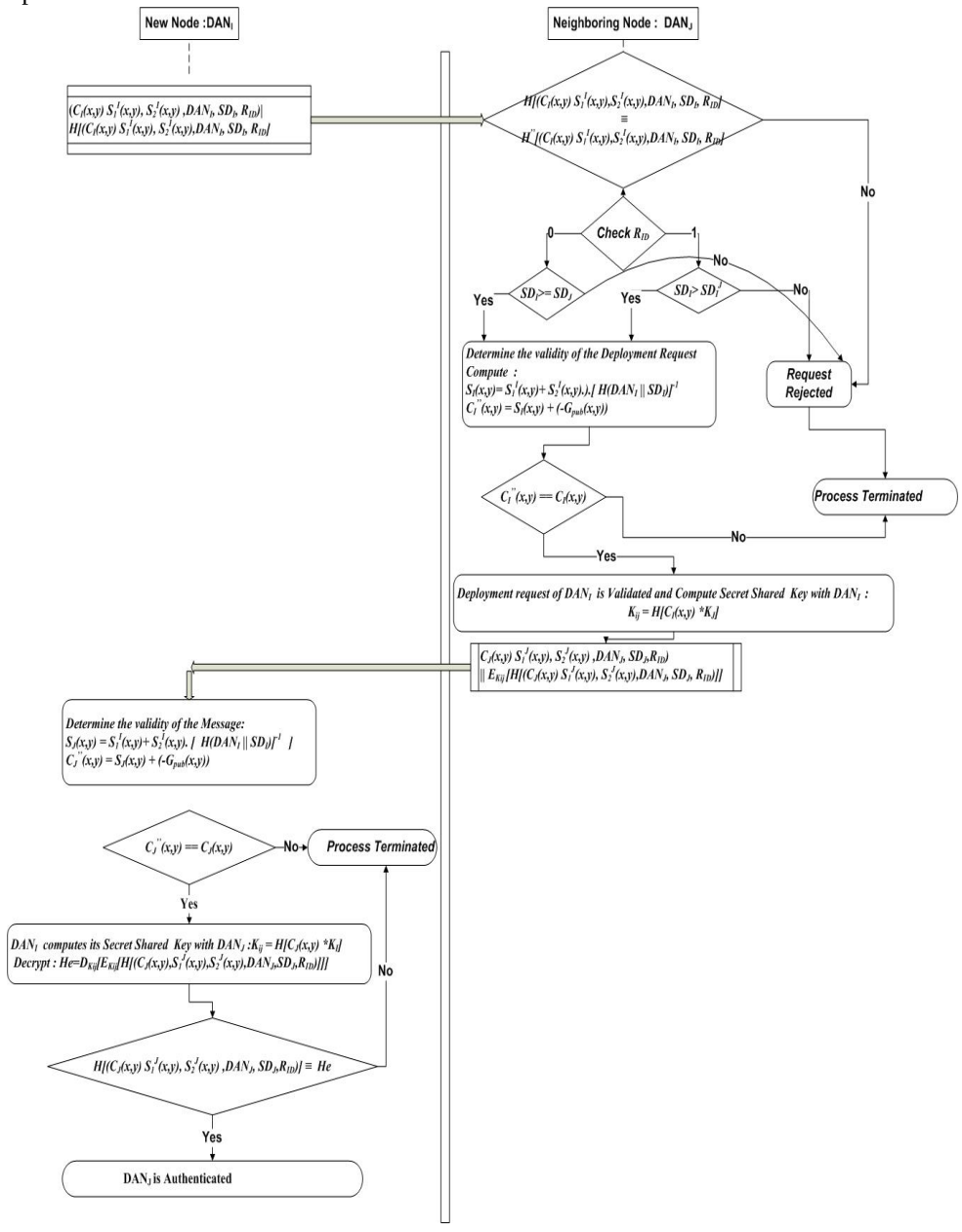


Figure 3. Flow Chart of the Proposed Access Control Scheme

2. **Session-Key-Reveal (SKR)**: defines a session key that is established between the two communicating roles.
3. **Weak Agree**: it is the weakest form of authentication claim. it determines that the communicating parties have been interacting at some point in time not necessarily in the present run
4. **Alive:** it signifies that the communicating parties are interacting with the intended partners.
5. **NI_Agree:** it is a more potent form of authentication. It determines whether the communicating parties agree with values exchanged.
6. **NI_Synch:** specifies whether the communicating parties are in synchronization or not by determining the messages sent and received are in the expected order.

More details about Scyther are given in [17]. The DAN_I is modeled as a role DAN-I, as shown in Figure 4. DAN_J is modeled as role DAN-J, as shown in Figure 5. The simulation parameters set in scyther are shown in Figure 6. In order to unbounded verification of the protocol, no of runs is disabled and set to 0. The matching type parameter is et to find all types of flaws and attacks. Within the advanced parameters, search pruning is et to find the best attack in order to perform the detailed and thorough security validation of the proposed protocol. The security verification output of Scyther is shown in Figure 7. From Figure 7, it can be depicted that the protocol is verified against all major claims depicting that protocol is safe against various active and passive attacks, thus making it suitable for practical applications.

```
role DAN-I
        {

        secret Sp,Sp1,Sp2;
        fresh Ki,DANi, SDi,Rid,Ki1,Ki2,G: Nonce;
        var  Kr,DANr, SDr,Rid,Kr1,Kr2,G: Nonce;

        //DANI sending : (CI(x,y) S1I(x,y), S2I(x,y) ,DANI, SDI, RID)||  H[(CI(x,y) S1I(x,y), S2I(x,y),DANI, SDI, RID]

        send_1( DAN-I, R, MUL(Ki,G) , MUL(Sp1,Ki1, H(DANi,SDi),G), MUL(Sp2,Ki2,
        H(DANi,SDi),G), DANi, SDi, Rid, H(MUL(Ki,G) , MUL(Sp1,Ki1, H(DANi,SDi),G),
        MUL(Sp2,Ki2, H(DANi,SDi),G), DANi, SDi, Rid ));

        //DANI  receives : (CJ(x,y) S1J(x,y), S2J(x,y) ,DANJ, SDJ,RID) || E Kij [H[(CJ(x,y) S1J(x,y), S2J(x,y),DANJ, SDJ, RID)]]

        recv_2(R, DAN-I, MUL(Kr,G) , MUL(Sp1,Kr1, H(DANr,SDr),G), MUL(Sp2,Kr2,
        H(DANr,SDr),G), DANr, SDr, Rid,{H(MUL(Kr,G) , MUL(Sp1,Kr1, H(DANr,SDr),G),
        MUL(Sp2,Kr2, H(DANr,SDr),G), DANr, SDr, Rid)}H(MUL( Ki,Kr,G) ));

        claim_i1(DAN-I,Secret,Ki);
        claim_i2( DAN-I, Alive );
        claim_i3( DAN-I, Weakagree );
        claim_i4( DAN-I, SKR, H(MUL( Ki,Kr,G) ) );
        claim_i5(DAN-I,Niagree);
        claim_i6(DAN-I,Nisynch);


        }
```

Figure 4. Role Specification of DAN_I

```
role DAN-J
        {

         secret Sp,Sp1,Sp2;
         fresh Kr,DANr, SDr,Rid,Kr1,Kr2,G: Nonce;
         var Ki,DANi, SDi,Rid,Ki1,Ki2,G: Nonce;

        //DANJ receives: (CI(x,y) S1I(x,y), S2I(x,y) ,DANI, SDI, RID)|| H[(CI(x,y) S1I(x,y), S2I(x,y),DANI, SDI, RID]

        recv_1(I, DAN-J, MUL(Ki,G) , MUL(Sp1,Ki1, H(DANi,SDi),G), MUL(Sp2,Ki2,
        H(DANi,SDi),G), DANi, SDi, Rid,H(MUL(Ki,G) , MUL(Sp1,Ki1, H(DANi,SDi),G),
        MUL(Sp2,Ki2, H(DANi,SDi),G), DANi, SDi, Rid));

        //DANJ sends : (CJ(x,y) S1J(x,y), S2J(x,y) ,DANJ, SDJ,RID) || E Kij [H[(CJ(x,y) S1J(x,y), S2J(x,y),DANJ, SDJ, RID

        send_2( DAN-J, I, MUL(Kr,G) , MUL(Sp1,Kr1, H(DANr,SDr),G), MUL(Sp2,Kr2,
        H(DANr,SDr),G), DANr, SDr, Rid,{H(MUL(Kr,G) , MUL(Sp1,Kr1, H(DANr,SDr),G),
        MUL(Sp2,Kr2, H(DANr,SDr),G), DANr, SDr, Rid)}H(MUL( Ki,Kr,G) ));

        claim_r1(DAN-J,Secret,Kr);
        claim_r2( DAN-J, Alive );
        claim_r3( DAN-J, Weakagree );
        claim_r4( DAN-J, SKDAN-J,H(MUL( Ki,Kr,G) ) );
        claim_r5(DAN-J,Niagree);
        claim_r6(DAN-J,Nisynch);


         }
```

Figure 5. Role Specification of DAN_J

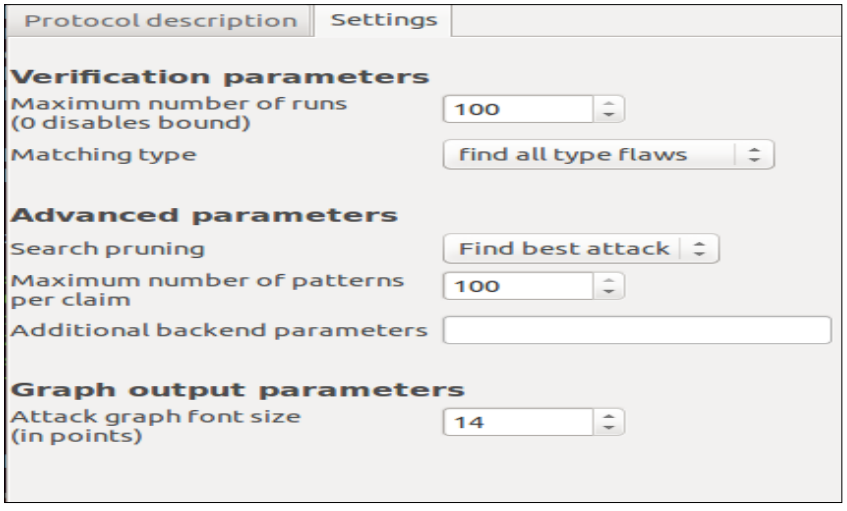


Figure 6. Simulation parameters in Scyther

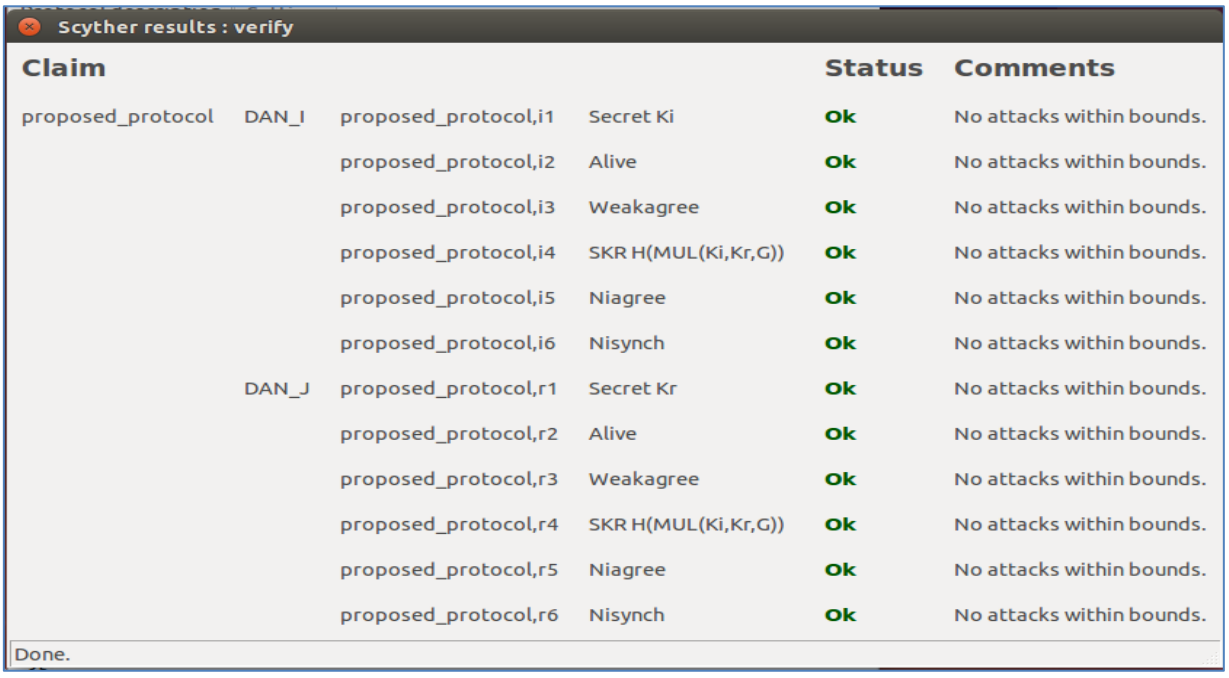


Figure 7.Verification Result in Scyther

6. COMPARISON WITH OTHER SCHEMES

A. *Computational; Communication and Memory overhead*

The computational overhead is analyzed based on the number of critical computational operations in the scheme. The various critical operations considered include T_{ESM}: Time taken for Scalar Multiplication; T_{INV}: Time taken for Modular Inverse; T_{HA}: Time is taken for Hash function, T_{ECE}: Time is taken for ECC Encryption; T_{DCE}: Time is taken for ECC Decryption; T_E: Time taken for symmetric key encryption; T_D: Time is taken for symmetric key decryption, T_{EPM}: Time taken for Point Addition. The most expensive operation among all of them is T_{ECE}, T_{DCE} and T_{ESM}. The total number of critical operations involved in the proposed scheme are: $2T_{ESM} + 3T_{HA} + 2T_{EPM} + T_{INV} + T_E/T_D$.

For estimating and comparing the communication and memory overhead, no of bits transmitted/received and no of bits to be stored at pre-deployment needs to be evaluated. The size of the various parameters involved in the proposed scheme is given in Table 3. The two messages exchanged in the protocol include;

1. $DAN_I \rightarrow * : (C_I(x,y)\ S_1^I(x,y),\ S_2^I(x,y)\ ,DAN_I,\ SD_I,\ R_{ID})\ ||H[(C_I(x,y)\ S_1^I(x,y),\ S_2^I(x,y),DAN_I,\ SD_I,\ R_{ID}]$
2. $DAN_I \rightarrow DAN_J : (C_J(x,y)\ S_1^J(x,y),\ S_2^J(x,y)\ ,DAN_J,\ SD_J,R_{ID})||\ E_{Kij}[H[(C_J(x,y)\ S_1^J(x,y),\ S_2^J(x,y),DAN_J,\ SD_J,\ R_{ID})]]$

The total no of bits transmitted in 02 messages is 2274 bits. Thus the total no of bits transmitted /received is 4548 bits. Memory overhead is computed by estimating the no of bits required to be stored at pre-deployed. Each DAN_I in the network is preloaded with the following information: 1. Elliptical Curve E(a,b). 2. Split Signature Pair (S_1^I, S_1^2) and $C_I(x,y)$. 3. Public key of the GWN: $G_{pub}(x,y)$. 4. Data Acquisition Node Identity: DAN_I. 5. Set Deployment Identifier: SD_I. Thus, The total of bits required to be stored at pre-deployment is 1632.

The comparative analysis of the proposed scheme in terms of computational, communication, and memory overheads as compared to the relevant existing schemes is shown in Table 4. Zhou et al. [24] have the highest overheads in terms of Computation, communication and memory, whereas Chatterjee et al. [33] have the lowest overheads. However, Chatterjee et al.[33] have significant limitations in terms of scalability and interdependence on time synchronization. The proposed scheme has moderate requirements in terms of computational, communication and memory overheads; however, it fulfills all important requirements needed for practical consideration of an access control scheme for a WSN based data acquisition tier.

B. *Security Comparison*

The security comparison is carried out based on significant security attacks, which a practical access control scheme must resist. The various attacks considered for comparison include False report Injection, Node capture. Sybil Attack, Worm Hole Attack Man in

the Middle Attack and Message Replay attack[13]. The description of the attacks[36], their general mitigation strategy[37] and how the proposed success control scheme mitigates them is formulated in Table 5. Besides that, the comparison is drawn on whether the scheme has been formally validated against various active and passive attacks using automated analysis. The comparison of the proposed scheme with existing schemes based on various security attacks and formal security validation criteria is tabulated in Table 6. From Table 6, it can be inferred that the proposed scheme is resistant to all significant attacks and has been formally verified against various active and passive attacks while fulfilling the important practical considerations of scalability and no interdependence on time synchronization.

C. Overall Comparison

The overall comparison of the proposed scheme with the relevant existing schemes is shown in Table 7. The proposed scheme with medium computational, communication and memory overhead fulfills the major requirement's which include scalability, no requirement of clock synchronization between the nodes in the network and formal verification against various active and passive attacks. The proposed scheme does not depend on GWN for new DAN addition. Each DAN is preloaded with the relevant information, which helps it to authenticate itself to its neighbors' in the communication range and thus limiting the intervention of GWN. The proposed scheme also does not require GWN to broadcast any new parameters after new node addition, as in the case of Haung[25] and Kim & Lee[26].

The design of the proposed scheme does not require clock synchronization within the network. At the time of deployment of the new DAN, the scheme uses the parameters SD_i and SD_i^j in order to distinguish between the old and a new node. The size of each of these parameters is 16 bit, thus providing a Re-deployment window of 2^{16} times for each node, which is fairly very large and practical. As such, the proposed scheme does not require timestamps in order to distinguish between the old and new nodes. The proposed scheme has also been validated formally against various active and passive attacks. From Table 7, it can be depicted that the proposed scheme provides a better tradeoff in terms of functional requirements and overheads as compared to the other existing schemes

Table 3. Size of Parameters

Parameter	**Size(bits)**
R_{ID}	1
DAN_I	16
SD_I	16
SD_i^j	16
$G_{pub}(x,y)$	320
$C_I(x,y)$	320
E(a,b)	320
H()	160
E_K	128
G(x,y)	320

Table 4. Comparison of Computational, Communication and Storage Overhead

Scheme	Computational Overhead	Communication Overhead Total No of Bits	Storage Overhead Total No of Bits
Zhou et al. (2007)[24]	$3T_{ESM} + T_{INV} + T_{HA} + 2\ T_{ECE}/\ T_{DCE}$	9152	1824
Huang (2009)[25]	$2T_{ESM} + 4T_{HA}$	3328 + 160*n	1456
Kim and Lee (2009)[26]	$2T_{ESM} + 9T_{HA}$	3328+512*n	1616
Huang (2011)[29]	$5T_{ESM} + 4T_{HA}$	3456	1648
Das et al. (2013)[31]	$4T_{ESM} + T_{INV} + 4T_{HA} + T_E/T_D$	4224	1560
Chatterjee et al.(2015)[33]	$8T_{HA} + T_E/T_D$	1800	896
Chatterjee et al.(2018)[13]	$2T_{ESM} + 5T_{HA} + T_E/T_D + T_{EPM}$	4288	1664
Proposed Scheme	$2T_{ESM} + 3T_{HA} + 2T_{EPM} + T_{INV} + T_E/T_D$	4612	1632

Table 5. Attacks and there Mitigation in the Proposed Access Control Scheme

Attack	Description	General Mitigation Mechanism	Mitigation in the Proposed Scheme
A1. False report Injection	An attacker can eavesdrop on the communication between the nodes in the WSN and inject false or masqueraded sensor data in the network.	It can be mitigated by establishing a shared key between the nodes in the network. The key established can be used for achieving confidentiality, and authentication sensed data transmitted between the nodes.	In the proposed access control mechanism, a shared key is established between the new node and its neighboring nodes. For a new node DAN_I and the neighboring node DAN_J, $K_{ij} = H[C_I(x,y) * K_J]$ is the shared key established, which can be used with any lightweight symmetric cipher to provide confidentiality and authentication of sensed data.
A2. Node capture	An attacker can capture a node and extract vital information from it, which can compromise the security of the complete network.	It can be mitigated by deploying the information within a node in such a way that even if Z nodes are captured in a network of N nodes, the security of N-Z nodes is not compromised.	The split signature pair (S_1^I, S_1^2) uses the private key of the base station. Due to the computational hardness of ECDLP, the private key of the BS cannot be extracted, thus preventing the comprise of complete network security.
A3. Sybil Attack	An Attacker can deploy a malicious node in a network that is capable of taking multiple identities, thus disrupting the network operation.	It can be mitigated by preventing the malicious node deployment in the network	The deployment request of a node is validated by evaluating the signature pair (S_1^I, S_1^2), as explained in Section 4. In order to deploy a malicious node, a forged signature pair needs to be created. The creation of a false signature pair is not possible as it requires access to the private key of the BS. Thus a malicious node cannot be deployed in the network
A4. Worm Hole Attack	An attacker can deploy a malicious node in a network that tunnels the packets to the distant neighbors and causes serve disruption in network routing		
A5. Man in the Middle Attack	An attacker establishes a forged pairwise key with the two legitimate nodes in a network and can eavesdrop, masquerade, and manipulate the traffic in the middle.	It can be mitigated by performing an authenticated key exchange between the node in the network	The symmetric key K_{IJ} between the DAN_I and DAN_J is formed in the DAN authentication and key exchange handshake, as explained in Section 4. The formation of the shared key is dependent on the authentication and validation of the signature pair of the nodes. Thus man in the middle attack is mitigated.
A6. Message Replay	An attacker can replay the old deployment request of a node to gain illegitimate access to the network.	It can be mitigated by guarding the freshness of the deployment request to distinguish between old and new deployments	The set deployment identifier is used to guard the freshness of the deployment as explained in section

Table 6. Security Comparison

Scheme	A1	A2	A3	A4	A5	A6	Formal Validation
Zhou et al. (2007)[24]	Yes	Yes	Yes	Yes	Yes	Yes	No
Huang (2009)[25]	No	No	Yes	Yes	No	No	No
Kim and Lee (2009)[26]	No	No	Yes	Yes	No	No	No
Huang (2011)[29]	No	No	Yes	Yes	No	No	No
Das et al. (2013)[31]	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Chatterjee et al. (2015)[33]							Yes
Chatterjee et al. (2018)[13]	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 7. Over All Comparison

Scheme	Resilient to Attacks?	Formal Validation	Scalability	Time Synchronization	Communication Overhead	Computation Overhead	Storage Overhead
Zhou et al. (2007)[24]	A1-A6	No	Yes	No	High	High	High
Huang (2009)[25]	A3, A4	No	No	Yes	Medium	High	Medium
Kim and Lee (2009)[26]	A3, A4	No	No	Yes	Medium	High	Medium
Huang (2011)[29]	A3, A4	No	Yes	No	High	Medium	Medium
Das et al. (2013)[31]	A1-A6	Yes	Yes	No	High	Medium	Medium
Chatterjee et al. (2015)[33]	A1-A6	Yes	No	No	Low	Low	Low
Chatterjee et al. (2018)[13]	A1-A6	Yes	Yes	No	Medium	Medium	Medium
Proposed Scheme	A1-A6	Yes	Yes	Yes	Medium	Medium	Medium

7. SIMULATION AND EXPERIMENTATION

A. *Experimental Setup*

The simulation and experimentation to perform a detailed energy analysis of the proposed access control protocol have been carried out on TinyOS[18] using the TOSSIM[19] simulator. TinyOS is an open-source operating system for developing WSN and IoT applications. TinyOS is based on a component model and event-driven programming. The component model helps in developing applications in a modular way, thus supporting reusability. Event-Driven programming helps in energy conservation as it involves the support for asynchronous or split phase operations. In TinyOS, an application is built by wiring various components together required for implementing the application logic. Components are of 2 types: Configuration and Modules. The configuration specifies the wiring of the various components that make the application and module implement the functional logic of the application. Each component is accessed by an interface wherein an interface declares the service in terms of commands and corresponding events to support split-phase functionality. A component providing an interface implements all its commands, and the component using the service implements all its events. The language used in TinyOS is NesC. The networking stack within TinyOS is shown in Figure 8. At the application layer, the NesC application targeting the TinyOS platform is designed and written. As there are multiple services on a WSN mote that needs to communicate through the same radio stack, an Active Message (AM) layer is provided to multiplex the access to the radio. Each service is identified using am "AM type." From the AM layer, the TinyOS application packet is passed down to the MAC layer and according transmitted over the wireless medium. In the implementation of the proposed access control protocol, the access control messages were given AM Type 19.

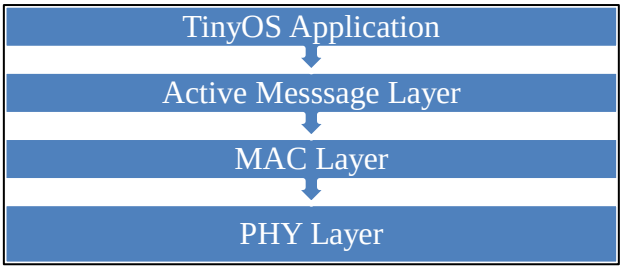


Figure 8. Networking Stack in TinyOS

TOSSIM is a discrete event simulator to simulate IoT/WSN applications developed in NesC language on the TinyOS platform. TOSSIM offers real deployment based signal propagation and noise models, thus providing a realistic simulation of IoT applications based on TinyOS. TinyViz is a Java visualization environment of the TOSSIM simulator. More details on TinyOS and TOSSIM can be found at [18][19].

The elliptical curve cryptography operations are implemented on TinyOS using TinyECC[38] library. TinyECC is a highly efficient implementation of ECC operations in NesC targeting the TinyOS platform. TinyECC provides various optimizations as given below:

1. **Barrett reduction:** is an efficient way to perform large integer reductions.
2. **Projective Coordinate Systems** is an alternate representation of a point of an elliptical curve in the form of (x,y,z), which results in the faster execution of point doubling and point addition.

3. **Sliding Window Method** is an optimization for speeding up scalar multiplication.

4. **Hybrid Multiplication:** is an efficient way to perform large integer multiplication. It maximizes the utilization of registers.

The description of the components used in implementing the proposed access control scheme is shown in un Table 8. The component graph of the configuration module developed is shown in Figure 9. The overall experimentation and simulation parameters are shown in Table 9. The radio model chosen for the simulation is Lossy. The lossy radio model places the nodes in a network as a directed graph. An edge (p,q) in the graph specifies that a signal from node p can be heard by the node q. Each edge has a value that denotes the bit error rate. The loosy radio model used in the application has been created using lossy builder, an inbuilt application in TinyOS. The number of nodes chosen in the simulation, as depicted in Table 8, is 8. The Bit Error Rate (BER) between the nodes in the simulation is shown in Table 10.

Table 8. Module Description

Module Name	Description
DANM	It provides the implementation logic of the proposed access control scheme
Main	A necessary component in every TinyOS application form where the execution starts
LedC	It provides the implementation for the control of Leds.
Generic Comm	It provides an implementation of generic communication operations in an Asynchronous Manner.
NNM	Provides the implementation for various number theory operations
ECCC	It provides an implementation of various ECC operations, which include point multiplication, point addition, scalar multiplication, etc.
SHA1M	It provides an implementation of SHA1 160 digest.
TimerC	It provides the implementation of timing control.
Secp160r1	It provides an implementation of the NIST curve SECP 160r1.

The energy overhead of the proposed protocol is estimated using PowerTOSSIM[19] PowerTOSSIM involves the usage of a component called Power State. During the simulation, the Power State component tracks the power state change of each simulated mote and logs it in a trace file. The trace file generated by the PowerState component is evaluated against an energy model to give an estimation of energy consumed by various hardware components of a mote. The energy model used in the simulation is a Mica2 model, as depicted in Table 11.

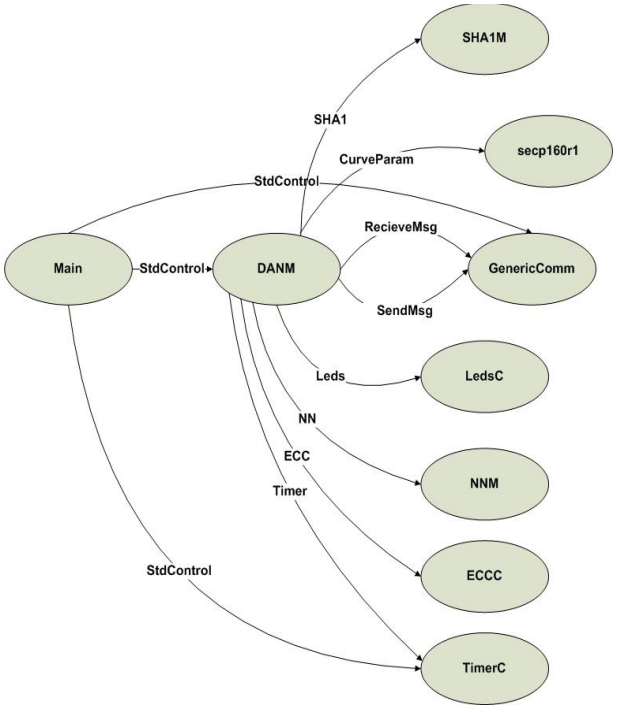


Figure 9. Component Graph of the proposed protocol

Table 9. TOSSIM Simulator parameters

S.No	Parameter	Description
1	Topology	Random
2	Grid size	5*5
3	Spacing Factor	10 ft
4	Curve	Secp160r1
5	Radio Model	Lossy Model
6	Optimization	Barrett reduction. Projective Coordinate Systems: Sliding Window Method: Hybrid Multiplication:
9	Channel	13
10	AM Type	19
11	Number of Nodes	8

Table 10. Bit Error Rate

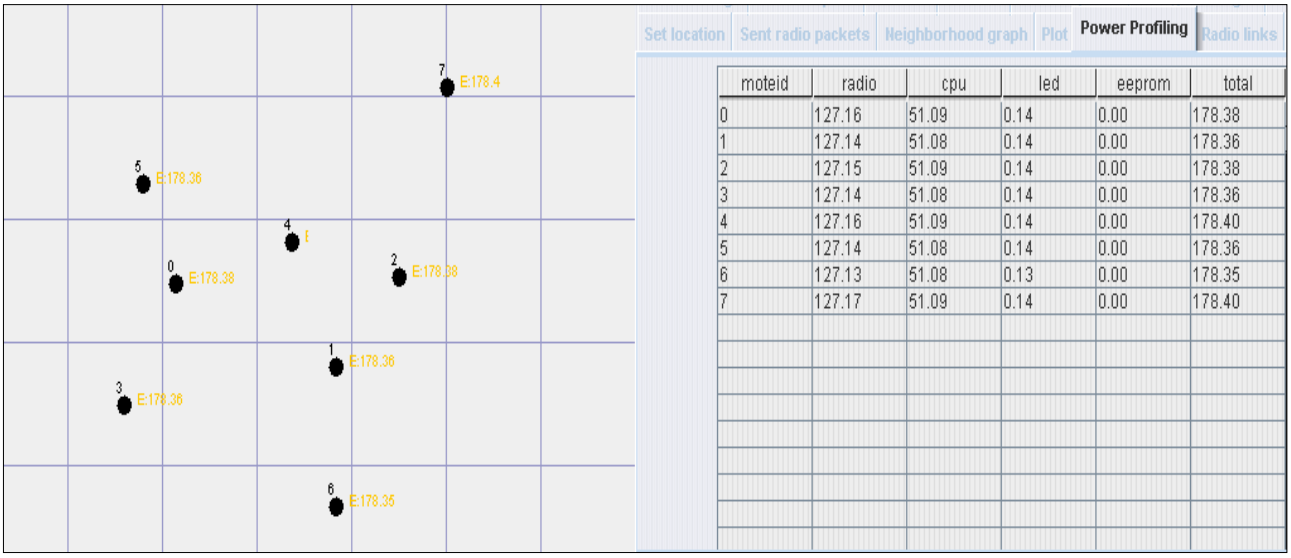
Node ID	BER with Neighboring Nodes						
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	0.002653	0.5	0.033279	0.047195	0.002147	0.013959	0.019745
1	**0**	**2**	**3**	**4**	**5**	**6**	**7**
	0.001576	0.002147	0.015508	0.029419	0.021707	0.002653	0.008325
2	**0**	**1**	**3**	**4**	**5**	**6**	**7**
	0.013198	0.002147	0.0	0.5	0.027122	0.022783	0.002653
3	**0**	**1**	**2**	**4**	**5**	**6**	**7**
	0.030079	0.016038	0.003114	0.001576	0.029419	0.5	0.017964
4	**0**	**1**	**2**	**3**	**5**	**6**	**7**
	0.047195	0.022416	0.006322	0.003114	0.5	0.5	0.008052
5	**0**	**1**	**2**	**3**	**4**	**6**	**7**
	0.002653	0.011436	0.5	0.033279	0.5	0.002147	0.002653
6	**0**	**1**	**2**	**3**	**4**	**5**	**7**
	0.012694	0.026128	0.005034	0.014728	0.034286	0.003541	0.002147
7	**0**	**1**	**2**	**3**	**4**	**5**	**6**
	0.003114	0.023545	0.001576	0.013198	0.020058	0.006624	0.002653

Table 11. Energy Model

CPU		**Radio**		**LED/Sensor Board/EEPROM**	
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	Read	6.2mA
Power Save	110 μA	Tx(-8 dBm)	6.5 mA		
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

B. Experimental Results

The simulation output on TinyViz is shown in Figure 10. 8 nodes have been deployed with mote id [0,1,2,3,4,5,6,7], respectively. As discussed in the experimental setup, these motes are randomly deployed with a spacing factor of 5 ft. Each node is fused with the NesC implementation of the proposed protocol. The PowerTOSSIM plugin of TinyViz depicts the energy consumed in terms of Radio, CPU, LED, and EEPROM. The EEPROM consumption is zero for each node as no operation using EEPROM was included in the developed application. LEDs were used for debugging, for signaling the sensing and the receiving of the access control messages and subsequently indicating whether a node is validated or not. On average, the energy consumed for CPU, radio, and LEDs is 127 mJ, 51 mJ, and 14 mJ, respectively, for each node. Thus the total average energy consumed for radio and CPU for each node is 178mJ. As indicated in [15], the energy results obtained have an excellent efficiency with an average error rate of 4% as compared to actual consumption.



moteid	radio	cpu	led	eeprom	total
0	127.16	51.09	0.14	0.00	178.38
1	127.14	51.08	0.14	0.00	178.36
2	127.15	51.09	0.14	0.00	178.38
3	127.14	51.08	0.14	0.00	178.36
4	127.16	51.09	0.14	0.00	178.40
5	127.14	51.08	0.14	0.00	178.36
6	127.13	51.08	0.13	0.00	178.35
7	127.17	51.09	0.14	0.00	178.40

Figure 10. Energy Consumed as indicated on Power Profiling plugin of TinyViz

CONCLUSION

In this paper, a secure and practical access control mechanism for new node deployment in the WSN based data acquisition tier of smart city applications is presented. The proposed scheme is based on elliptical curve cryptography and addresses the practical concerns of scalability and no interdependence on clock synchronization between the nodes in the network. A detailed comparative analysis of the proposed scheme with the relevant existing scheme has been presented. The comparison indicates a better tradeoff as compared to the relevant existing schemes. The scyther simulation of the scheme indicates that the proposed protocol is safe against various active and passive attacks. The scheme has also been simulated on TinyOS using TOSSIM to carry out a detailed energy analysis in reference to a practical energy model. The energy analysis suggests a node requires 178 mJ in total to implement and execute the proposed access control protocol. Future advancement in the proposed work would be to incorporate the mechanism of user access control. A user access control mechanism would allow a legitimate user to query the data acquisition tier for specific information, thus enhancing the secure interactivity within IoT based smart applications.

REFERENCES

[1] Hussein T. M, Melike E.K and Mubashir H R," Wireless Sensor Networks in Smart Cities: Applications of Channel Bonding to Meet Data Communication Requirements," Transportation and Power Grid in Smart Cities: Communication Networks and Services, Wiley, pp.247-268, 2009

[2] Rajab, Husam & Cinkler, Tibor," IoT based Smart Cities." 10.1109/ISNCC.2018.8530997,2018.

[3] Erdem, Ahmet, Sevgi Özkan Yildirim, and Pelin Angin. "Blockchain for Ensuring Security, Privacy, and Trust in IoT Environments: The State of the Art." In Security, Privacy, and Trust in the IoT Environment, pp. 97-122. Springer, Cham, 2019.

[4] A.J. Moshayedi, M. S. Hosseini, and F. Rezaee, "WiFi Based Massager Device with NodeMCU Through Arduino Interpreter," J. Simul. Anal. Nov. Technol. Mech. Eng., vol. 11, no. 1, pp. 73–79, 2019.

[5] Wireless Sensor Networks Market by Offering (Hardware, Software, Services), Sensor Type, Connectivity Type, End-user Industry (Building Automation, Wearable Devices, Healthcare, Automotive & Transportation, Industrial), and Region - Global Forecast to 2023. [Online]. Available:https://www.marketsandmarkets.com/MarketReports/wireless-sensor-networks-market-445.htmlet-of-thingsstatistics/. Accessed on February 20, 2020.

[6] A. H. Moon, U. Iqbal, and G. M. Bhat," Secured data acquisition system for smart water applications using WSN." Indian Journal of Science and Technology 9, 10, pp.1—11,2015

[7] Abdmeziem M.R., Tandjaoui D., Romdhani I., "Architecting the Internet of Things: State of the Art," Koubaa A., Shakshuki E. (eds) Robots and Sensor Clouds. Studies in Systems, Decision and Control, vol 36. Springer, Cham,2016.

[8] Mote Works, "Getting Started Guide" PN: 7430-0102-02,2013

[9] Gura, Nils & Patel, Arun & Wander, Arvinderpal & Eberle, Hans & Shantz, Sheueling.” Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs”. Lect Notes Comput Sci. 3156. 119-132. 10.1007/978-3-540-28632-5_9, 2004.

[10] Malan, David & Smith, Michael & Welsh, Matt, “Implementing Public-Key Infrastructure for Sensor Networks,” ACM Transactions on Sensor Networks. 4. 10.1145/1387663.1387668,2008.

[11] Parno, B., Perrig, A. and Gligor, V., “Distributed detection of node replication attacks in sensor networks’, Proceedings of the IEEE Symposium on Security and Privacy (S&P’05), pp.49–63,2005.

[12] Hu Y.C, Perrig.A and Johnson. D.B.,” Wormhole Attacks in Wireless Networks,” IEEE Journal on Selected Areas in Communication, Vol. 24, No. 2, pp. 370 – 380,2006.

[13] Chatterjee, S. and Roy, S.,” An efficient dynamic access control scheme for distributed wireless sensor networks,” Int. J. Ad Hoc and Ubiquitous Computing, Vol. 27, No. 1, pp.1–18,2018.

[14] Zheng, Jun & Jamalipour, Abbas., ”Wireless Sensor Networks: ANetworkingPerspective”.10.1002/9780470443521,2009.

[15] Carman, David & Kruus, Peter & Matt, Brian,” Constraints and approaches for distributed sensor network security” 2000.

[16] S. M. Lasassmeh and J. M. Conrad, "Time synchronization in wireless sensor networks: A survey," Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon), Concord, NC, 2010, pp. 242-245

[17] Cremers, Cas.,” The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols.” 5123. 414-418. 10.1007/978-3-540-70545-1_38,2008.

[18] Levis P and Gay D,” TinyOS Programming,” Cambridge University Press,2008

[19] Shnayder, Victor & Hempstead, Mark & Chen, Bor-rong & Werner-Allen, Geoffrey & Welsh, Matt.” Simulating the Power Consumption of Large-Scale Sensor Network Applications”, SenSys'04 - Proceedings of the Second International Conference on Embedded Networked Sensor Systems. 188-200. 10.1145/1031495.1031518.,2004.

[20]Koblitz N, “Elliptic curve cryptosystems,” Math Comput 48:203–209,1987.

[21]Miller V,” Use of elliptic curves in cryptography,” Advances in Cryptology—CRYPTO ’85, Lecture Notes in Computer Science, vol 218. Springer, 1986, pp 417–426.

[22] Hankerson, Darrel & Menezes, Alfred & Springer, Scott. ,” Guide to Elliptic Curve Cryptography.” 332. 10.1007/978-1-4419-5906-5_245,2004.

[23]Hu, X.; Zheng, X.; Zhang, S.; Li, W.; Cai, S.; Xiong, X. A High-Performance Elliptic Curve Cryptographic Processor of SM2 over GF(p). *Electronics* 2019, *8*, 431

[24] Y. Zhou, Y. Zhang and Y. Fang, “Access control in wireless sensor networks,” Ad Hoc Networks, vol. 5, no. 1, pp. 3–13, 2007.

[25] H.-F. Huang, “A novel access control protocol for secure sensor networks,” Computer Standards and Interfaces, vol. 31, no. 2, pp. 272–276, 2009.

[26] H.-S. Kim and S.-W. Lee, “Enhanced novel access control protocol over wireless sensor networks,” IEEE Transactions on Consumer Electronics, vol. 55, no. 2, pp. 492–498, 2009.

[27] P. Zeng, K.-K. R. Choo, and D.-Z. Sun, “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, 2010.

[28] J. Shen, M. Sangman, and C. Ilyong, “Comment: enhanced novel ACP over wireless sensor networks,” IEEE Transactions on Consumer Electronics, vol. 56, no. 3, pp. 2019–2021, 2010.

[29] Huang, H-F,” A new design of access control in wireless sensor networks,” International Journal of Distributed Sensor Networks, Article ID 412146, p.7,2011.

[30] H. Lee, K. Shin, and D. H. Lee, “PACPs: practical access control protocols for wireless sensor networks,” IEEE Transactions on Consumer Electronics, vol. 58, no. 2, pp. 491–499, 2012.

[31] Das, A.K., Chatterjee, S. and Sing, J.K,” A novel efficient access control scheme for large-scale distributed wireless sensor networks,” International Journal of Foundations of Computer Science, World Scientific Publishing Company, Vol. 24, No, 5, pp.625–653,2013.

[32] Chatterjee, S., Das, A.K. and Sing, J.K.,” An enhanced access control scheme in wireless sensor networks,” Ad Hoc and Sensor Wireless Networks, Vol. 21, No. 1, pp.121–149.2014

[33] Chatterjee, S., Das, A.K. and Sing, J.K. (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.

[34]Dolev, D. and Yao., A., “On the security of public-key protocols,” Proceedings of the IEEE 22nd Annual Symposium on Foundations of Computer Science, Nashville, TN, USA, pp.350–357,1987.

[35] Dolev, D. and Yao, A.,” On the security of public-key protocols,” IEEE Transactions on Information Theory, Vol. 29, No. 2, pp.198–208,1983

[36] Dhillon, Parveen, and Manpreet Singh. "Internet of Things Attacks and Countermeasure Access Control Techniques: A Review." International Journal of Applied Engineering Research 14, no. 7 (2019): 1689-1698.

[37] Suciu, G., Ijaz, H., & Pastea, D. V.,” The IoT Devices and Secured Communication Architecture and Use Cases” 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) (pp. 1-5). IEEE.2019

[38] Liu and P. Ning, “TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks,” International Conference on Information Processing in Sensor Networks (ipsn 2008), St. Louis, MO, 2008, pp. 245-256

Ummer Iqbal received the 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, Srinagar in 2010 as Scientist 'C ' at NIELIT Srinagar/Jammu. his MTech in Communication and Information Technology (CIT) from the National Institute of Technology Srinagar, India in 2015. He is currently pursuing his Ph.D. in Wireless Sensor Networks at the National Institute of Technology (NIT), Srinagar, India. His research & development interest is in wireless sensor networks, Network Security and information security & open source technologies.

Ajaz Hussain Mir has done his Bachelor of Engineering (B.E) in Electrical Engineering with specialization in Electronics & Communication Engineering (ECE). He did his Master of Technology (M.Tech) in Computer Technology and Ph.D. both from IIT Delhi in the year 1989 and 1996, respectively. He is Chief Investigator of Ministry of Communication and Information Technology, Govt. of India project: Information Security Education and Awareness (ISEA). Presently, he is a Professor in the Department of Electronics & Communication Engineering at NIT Srinagar, India. He has been guiding Ph.D. and M.Tech thesis in security and other related areas and has many international publications to his credit. His areas of interest are biometrics, image processing, security, wireless communication, and networks