Modeling and performance evaluation of elliptical curve digital signature algorithm–based secure clustering and symmetric key establishment in heterogeneous wireless sensor networks

Agbu, Alexander Uchenna[1]
Department / Office of National Space Research and Development Agency, (NASRDA)
Abuja, Nigeria
agbualexuche@gmail.com

Ozuomba Simeon[2]
Department Of Electrical/Electronic And Computer Engineering,
University of Uyo, Akwa Ibom State Nigeria
simeonoz@yahoo.com
simeonozuomba@uniuyo.edu.ng

Philip M. Asuquo[3]
Department Of Electrical/Electronic And Computer Engineering,
University of Uyo, Akwa Ibom State Nigeria
philipasuquo@uniuyo.edu.ng

***Abstract*— In this paper, modeling and performance evaluation of elliptical curve digital signature algorithm (ECDSA)–based secure clustering and symmetric key establishment in heterogeneous wireless sensor Networks (HWSNs) are presented. The symmetric key management scheme incorporates pairwise keys for secure communication among sensor nodes in the heterogeneous WSNs. The network model along with explanation regarding secure clustering and symmetric key establishment in the HWSNs are presented along with elaboration on how security is established in the initial phase of bootstrapping and clustering of these networks. Relevant mathematical models pertaining to the proposed ECDSA scheme are presented and then the performance of the ECDSA key distribution scheme is compared with other existing and commonly used distribution techniques. The results show that while providing similar probability of key sharing among nodes, the ECDSA scheme significantly minimizes the storage requirements and better link compromise probability. The results also show that the ECDSA scheme requires lower number of hops, hence, minimizes the probability of compromise and also saves sensor nodes energy.**

Keywords*— *Elliptical Curve Digital Signature Algorithm, Secure Clustering, Survivable Routing, Symmetric Key Establishment, Heterogeneous Wireless Sensor Networks

1. Introduction

The substantial rise of wireless sensor networks (WSNs) utility in diverse applications such as hostile, unattended, and inaccessible environments mandates the users to be more assured about the security compared to the survivability. The intrinsic nature of wireless sensor nodes, such as being subject to resource constraints (power, processing, and communication), easily reproduced, and possibly tampered with, causes other security strategies developed for infrastructure based wireless networks to be infeasible for WSNs [1, 2]. A typical example of these sensor nodes is the reduced function devices (RFDs) which are defined in the IEEE 802.15.4-2006 standard [3].

In as much as security strategies provide confidentiality, integrity, and authentication, which are critical for such applications, a secure and survivable infrastructure is always desired. Network survivability is defined as the ability of the network to fulfill its mission in the presence of attacks and/or failures in a timely manner [4]. Being a typical criteria to enhance scalability and survivability in the WSNs, clustering sensor nodes into some groups has been considered in several literatures [5–9]. Sequel to the energy constraint nature of wireless sensor nodes and their limited transmission range, establishing multi-hop routing toward the gateway is more efficient than having direct transmission [7]. Besides, transmission of data consumes the most energy compared to data computation. As a result, sending signals in an optimal power level is very important.

From the security stand point, through compromising a sensor node by an adversary in a multi-hop path, the information on the node is revealed, and an attacker might be able to control the operation of the vulnerable node. Hence, for the purpose of providing security to communication links in WSNs, all messages should be encrypted and authenticated by any two individual sensor nodes engaged in message exchange [10].

Essentially, secure clustering and key establishments are exigent issues in the WSNs. Hence, an efficient key management scheme should be designed to share the

cryptographic keys amongst the sensor nodes. It is clear that using a single traditional symmetric key is not secure; since sensor nodes are not tamper proof and upon being caught by an attacker, all information will be exposed to the attacker [11]. In recent researches, incorporating pairwise keys for secure communication among sensor nodes in the heterogeneous WSNs has been considered [12, 13]. Consequently, in this research, the modeling and performance evaluation of elliptical curve digital signature algorithm–based secure clustering and symmetric key establishment in heterogeneous wireless sensor networks is presented. Furthermore, the performance of the elliptical curve digital signature algorithm is evaluated and compare with the performance of other secure clustering and symmetric key establishment approaches for heterogeneous wireless sensor networks.

2. Methodology

2.1 Network Model

This paper focuses on secure clustering and symmetric key establishment in Heterogeneous Wireless Sensor Networks (HWSNs) [14,15,16,17,18,19]. First, the network model (as shown in Figure 1) along with explanation regarding secure clustering and symmetric key establishment in Heterogeneous Wireless Sensor Networks are presented. Then, the paper elaborates on how security is established in the initial phase of bootstrapping and clustering of these networks. It is assumed in this model that the number of gateways is relatively very small compared to the number of sensor nodes, that is, $G \ll N$, and the gateways recognize their location data and can communicate securely with each other, and the base station (BS).

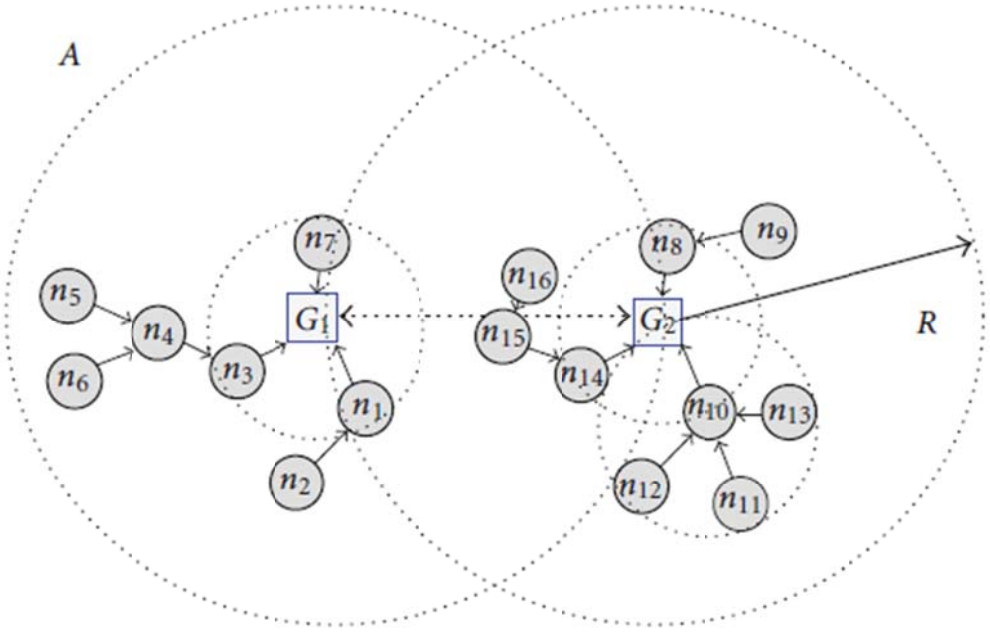

Figure 1: Clustered WSN with two gateways and sixteen sensor nodes deployed in area A

In order to meet the HWSN coverage requirements, it is assumed that all sensor nodes are distributed evenly and randomly in the monitoring area, A. Also, it is assumed that sensor nodes do not have any knowledge about their geographic location information. This model proposes two phases of operations, namely, preloading and deployment phases.

Let n_i and G_j represent the sensor node $i, i \in \{1, 2, \dots, N\}$ and gateway $j, j \in \{1, 2, \dots, G\}$, respectively, in the network. The assumption here is that each sensor node and gateway are identified by unique ID number i and j, respectively, where N and G represents the largest numbers. The number of edges connected securely to a sensor node n_i is denoted by $\deg n_i$. The ranges of transmission for all sensor nodes and all the gateways are denoted by r and R, respective, where $R > r$. Hence, communication between a sensor node and a gateway can be established if they are within the distance r of each other.

A set of sensor nodes N in this context is considered as a cladding set of area, A if and only if for each point say $P \in A$, there is $n_i \in N$ that n_i covers P. The sensor node n_i clads P if it is within the transmission range of n_i, which is r. The largest radius of a cluster is cladded by a gateway G_j, which is defined by R and approximated by multiplying the transmission range of each sensor node, r, with the number of hops to the gateway, h. This implies that $R_{G_j} = h \times r$.

Let a connected weighted graph $\mathcal{g} = (V, E)$, the minimum spanning tree covers all the vertices V (contains $|V| - 1$ edges) of $\mathcal{g}$ which has minimum total edge weight.

A spanning tree of $\mathcal{g}$, consisting of a root node s, such that the distance between s and all other vertices in $\mathcal{g}$ stays minimal is considered as the shortest path tree of a connected weighted graph $\mathcal{g}$. Achieving minimum weight is the goal of a minimum spanning, while distance

preservation from the root is the goal of the shortest path tree. Essentially, digital signature is a cryptographic tool and mathematical scheme for demonstrating nonrepudiation, authenticating the integrity and origin of a signed message. A private key is typically engaged by the signer to generate the digital signature for the message, and the public key is used by anyone to verify the signature.

Before sensor nodes are arbitrarily deployed in an environment, the required keys are generated and preloaded by the server based on elliptic curve cryptography (ECC) into the sensor nodes and gateways. As shown in Figure 2, a sensor node, $n_i, 1 \leq i \leq N$, is preloaded with its own public key, given as $P^r_{n_i}$, and the public key of all existing gateways in the network, given as $\left\{P^u_{G_j} \mid 1 \leq j \leq G\right\}$. As a result, the gateway G_j preloaded with the public key of all gateways in addition to its own $\left\{P^u_{G_j} \mid 1 \leq j \leq G\right\}$, its private key $P^r_{G_j}$ and public keys of all the sensor nodes $\left\{P^u_{n_i} \mid 1 \leq i \leq N\right\}$ in the network.

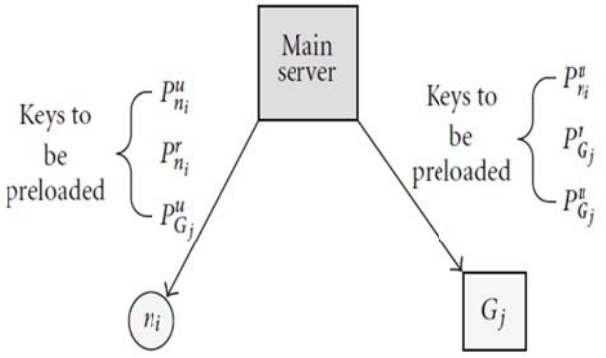

Figure 2: Embedded keys into gateways and sensor nodes

These keys are embedded in the gateways and sensor nodes. In clustered WSNs, sensor nodes are deployed randomly and evenly in a manner similar to distributed WSNs. The gateways are deployed within the field, such that each sensor node can communicate with at least one gateway. This is possible by varying the transmission range of gateways, R, in the network during the initial communication setup. It is assumed that the gateways are aware of the location of the base station (BS) and communicate with the BS directly or in a multi-hop manner securely.

2.2 The Proposed Secure Clustering Approach

In this paper, elliptical curve digital signature algorithm (ECDSA) is adopted for securing the symmetric key used in the clustered WSNs. Ideally, sensor nodes in clustered WSNs are securely partitioned into clusters. Therefore, it is assumed that if the attacker exists in the field, they are unable to comprehend the exchanged information. In Figure 1, a network with two gateways ($G_1$and G_2) and 16 sensor nodes (n_1 to n_{16}) is illustrated.

The gateway G_j in each cluster securely discovers all the sensor nodes which belong to it. Furthermore, sensor nodes are aware of their assigned gateway and cluster. As illustrated in Figure 3, each gateway G_j broadcasts the message B_{G_j} to all sensor nodes with random delay, that is,

$$G_j \rightarrow n_i$$

$$B_{G_j} = \left(ECDS_{P^r_{G_j}}\left\{h\left(M||ID_{G_j}\right)\right\}, P^u_{G_j}, M, ID_{G_j}\right) \quad (1)$$

Where, M represents the broadcast message, $h(\cdot)$ denotes the one-way hash function, $||$ denotes the concatenation operator. B_{G_j} is calculated by G_j. First of all, the one-way hash function $h(\cdot)$ is executed over the $\left(M||ID_{G_j}\right)$, the elliptical curve digital signature $ECDS$ is calculated over the hash results by using the private key of the gateway G_j, that is, $P^u_{G_j}$, message M and ID_{G_j}. To ensure that the maximum number of sensor nodes receives the broadcast, the broadcast is repeated several times.

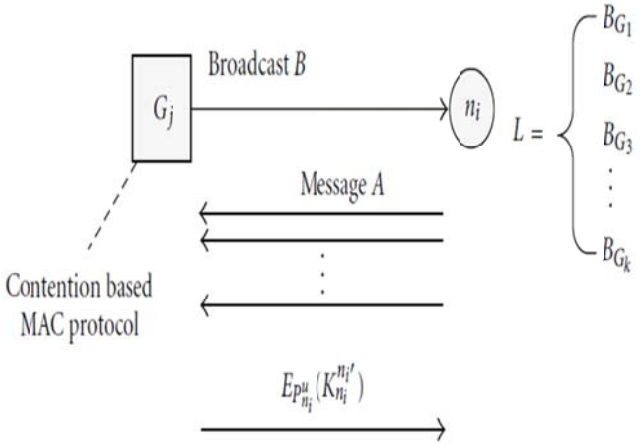

Figure 3: Information exchange between sensor nodes and gateways during secure clustering

For the purpose of message authentication, once the broadcast message is received, the sensor node n_i compiles the list of all the received messages from the gateways as $\ell = \left\{B_{G_1}, B_{G_2} \ldots, B_{G_k}\right\}$, where $k, 1 \leq k \leq G$ denotes the number of gateways from which a sensor node received a broadcast message. The priority of the generated list is a function of signal-to-noise ratio (SNR) of the received message, that is, $P_{B_{G_1}} > P_{B_{G_2}} > \cdots > P_{B_{G_k}}$ where $P_{B_{G_k}}$represents the received signal power from the gateway G_k for $1 \leq k \leq G$. Thereafter, each sensor node n_i verifies the message integrity using ECDSA with public key of the gateway and compares the received public key with the preloaded counterpart. Notably, verifying the authenticity of the public key of a gateway implies finding out whether the attached public key of the gateway matches the one embedded in the memory of a sensor node. If the received public key does not match the pre-loaded one, sensor node n_i will definitely reject the broadcasted message. This stops

sensor nodes from performing expensive verification on the fake signatures broadcasted from the attackers.

In addition, each sensor node n_i is capable of determining the distance d_{n_i} from the desired gateway G_j and this is achieved by introducing received signal strength indicator (RSSI). The minimum threshold distance from the gateway G_j is considered as one-hop distance, $d = \min\{d_{n_i}, 1 \leq i \leq N\}$, in which sensor nodes within this distance can communicate with the gateway directly.

The ECDSA algorithm is deployed at this point. This will be used by the gateway in each cluster to find which sensor nodes select the gateway G_j as their cluster head. The gateway G_j broadcasts a message requesting sensor nodes to inform the gateway if they are within the communication distance d from the gateway. In this scenario, each sensor node n_i encrypts its ID concatenated with its public key using the public key of the desired gateway. A sensor node transmits this message at maximum power to acknowledge the desired gateway in the top of its list ℓ as follows:

$$n_i \rightarrow G_j: A = E_{P_{G_j}^u}\left(ID_{n_i} || P_{n_i}^u\right) \quad (2)$$

Where, $E_{P_{G_j}^u}(\cdot)$ represents the encryption function using the public key of gateway, G_j. The gateway, G_j then decrypts this message by using it private key as:

$$G_j : D_{P_{G_j}^r}(A) = ID_{n_i} || P_{n_i}^u \quad (3)$$

In this scenario, the gateway G_j compares the received public key from the sensor nodes with the ones embedded in its memory prior to deployment. This occurs to prevent an attacker from throwing illegitimate nodes into a cluster and mounting a denial-of-service (DoS) attack.

As large number of sensor nodes respond to a gateway, avoiding contention is difficult. Since contention results in collisions, this affects the survivability of the network. Hence, a befitting medium access control (MAC) protocol is required to be installed in each sensor node. It should be noted that presuming sensor nodes to be time synchronized is not realistic because of the large number of nodes. To overcome this challenge, the contention-based and self-stabilizing MAC protocol is incorporated here. Eventually, each gateway will make a list of all the sensor nodes in its cluster along with their IDs and public keys.

Now, the public keys of the sensor nodes and gateways are authenticated. Hence, each gateway G_j will require its one-hop sensor nodes n_{1i} (e.g., n_8, n_9 $n_{10}0$, and n_{14} of cluster 2 ,as shown in Figure 1) within the cluster to broadcast a message to ask its one-hop neighbors in the cluster to report to n_{1i}. In this scenario, sensor node n_{1i} emerges as the parent node to the nodes in its one-hop neighborhood. In the same way, the other nearby nodes asks their one-hop neighbors to report themselves. Hence, every node within the cluster connects to the gateway in a single or multi-hop route, that is, $n_{1i}, n_{2i}, n_{3i}, \ldots, n_{hi}$, where h denotes the number of hops from a node n_i to the gateway G_j. These sensor nodes send their information to the n_{1i} node, which informs the gateways about these sensor nodes.

Every sensor node which has selected G_j as the gateway and is within the desired cluster will be discovered by the gateway G_j. It should be noted that a unique path exists from each node to the gateway as each node has just one parent. An appropriate routing algorithm is required to route the information to the gateway in each cluster. It defines the path that the packets can be pushed to get to the gateway. Hence, a minimum cost path algorithm can be used to find the optimized spanning tree rooted at the given node.

The nodes that directly follow the root Node n_i in the minimum cost tree are made up of the minimum neighborhood of node n_i. The minimum cost routes between the gateway G_j and the node n_i are all contained in the minimum neighborhoods of the nodes.

2.3 Secure and Survivable Routing

This section presents routing algorithm for the sensor nodes to forward data toward the gateway in each cluster. If data from neighborhoods are highly tallied, then the minimum spanning tree (MST) is profitable in terms of survivability and network lifetime. However, in the scenario where there is flow correlation amongst sensor nodes, shortest path tree (SPT) should be integrated to achieve survivability and better network lifetime. Furthermore, shorter paths are more secure than the longer paths (this will be expanded in subsequent sections). It should be noted that using the shortest path limits the number of paths that can be used to relay data toward the gateway.

The use of link estimation and parent selection (LEPS) scheme is deployed as a routing algorithm. In this scheme, each node watches all traffic received within the one-hop range, including route updates from the neighbor nodes. By applying the least cost path, it manages the nearest available neighbor node and decides the next hop. To locate a least cost path, one needs to compute the costs of all edges between each sensor node then obtain a set of least cost paths. To achieve this, a cost function defined in Equation 4 is applied. The following parameters are defined:

i. $f(E_{n_i})$: denote the function of residual energy of the sensor node n_i, $\forall\, i \in \{1, 2, \ldots, N\}$.

ii. $d_{n_i,n_{i'}}$: denote the distance between the sensor node n_i and $n_{i'}$

iii. $F(e_{n_i,n_{i'}})$: denote the error function between the sensor node n_i and $n_{i'}$.

Hence, the cost function for a link between sensor node n_i and $n_{i'}$ can be computed as:

$$C_{n_i,n_{i'}} = \left(d_{n_i,n_{i'}}\right)^{\alpha} + f\left(E_{n_i}\right) + F(e_{n_i,n_{i'}}) \quad (4)$$

Where α denotes the free space loss exponent and typically $\alpha \geq 2$. The error function relates to the maximum data buffered in sensor node b and the distance between sensor node n_i and $n_{i'}$. This can be written as:

$$F\left(e_{n_i,n_{i'}}\right) = c_0 \cdot \frac{d_{n_i,n_{i'}}}{b} \quad (5)$$

Where c_0 represents a constant coefficient. To obtain the least cost path from a sensor node n_i to the gateway G_j, the quantity of hops should be considered.

2.4 Symmetric Key Establishment

Once secure clustering is set up, broadcast authentication, and determining the desired routing algorithm among sensor nodes and gateways, sensor nodes establish secured communication between each other to access the gateway securely in a multi-hop path. Since gateways have knowledge of the one-hop neighbors of the sensor nodes and also have enough information to control the sensor nodes, they send pairwise keys to each sensor node and its potential one-hop neighbors. To accomplish this, gateway G_j will transmit the pairwise key to the sensor node n_i which is easily found between its neighbors n_i with regards to the least cost path routing algorithm.

Above all, the symmetric key generated for the sensor node n_i and $n_{i'}$, which is, $K_{n_i}^{n_{i'}}$, is encrypted using the public key of the sensor node n_i ,which is $E_{P_{n_i}^u}\left(K_{n_i}^{n_{i'}}\right)$, for $1 \leq i, i' \leq N$. After the generation and encryption of the symmetric key, each gateway G_j unicast this message to the sensor node n_i. Each sensor node decrypts this message via its own private key $P_{n_i}^r$ and gets the symmetric key $K_{n_i}^{n_{i'}}$. Since this message is encrypted by the public key (following the concept of ECC) of every sensor node, then revealing the symmetric key is not an easy task to the attacker. For instance, in Figure 1, the sensor node n_4 will get the symmetric keys for nodes n_3, n_5, n_6 as $K_{n_4}^{n_3}$, $K_{n_4}^{n_5}$, $K_{n_4}^{n_6}$, respectively.

2.5 Unicast Authentication

One pertinent issue to address is how sensor node n_i ensures that the encrypted symmetric key, which is $E_{P_{n_i}^u}\left(K_{n_i}^{n_{i'}}\right)$ originates from the gateway G_j and not from an attacker. A proposed solution to this is the use of the elliptical curve digital signature algorithm (ECDSA) authentication. To ensure that the message $E_{P_{n_i}^u}\left(K_{n_i}^{n_{i'}}\right)$ is unicasted from the gateway G_j, the elliptic curve digital signature can be computed by the gateway on the message. Hence, sensor node n_i can check the signature using the public key gateway G_j, and this guarantees that the message emanated from a legitimate gateway G_j, and not from an attacker. One of the requirements for this scheme is N times signature generation by the gateways, and all the sensor nodes have to verify and decrypt the unicasted message. Notably, the computation cost will increase since the verification of signature is expensive operation. However, some of the overheads can be reduced by a one-time signature. In that case, each sensor scheme and its corresponding gateway are allowed to get a shared symmetric key during the first broadcast authentication integrating the elliptic curve Diffie-Hellman (ECDH) method.

Then, in deploying symmetric key, the unicast authentication can be performed by generating a message authentication code (MAC). Hence, any unicast from the gateway can be authenticated by the sensor nodes. Authentication methods simply mean overheads in computation and communication times. Hence, there is need to strike a balance between the required level of security in the authentication and the costs (in terms of computation and communication times), else the arising overheads might be against the survivability of the network. Apart from giving guaranty for confidentiality and authentication, it is imperative to ensure that data is recent, and no attacker replayed old messages. A sensor node n_i can accomplish this by using nonce (which is an unpredictable random number). In the presented scheme, before unicasting the symmetric keys by the gateways, sensor node n_i can transmit a key request message to the gateway G_j accompanied with an arbitrary nonce, that is, N_{n_i} and encrypted by $P_{G_j}^u$. Hence, any time a gateway seeks to unicast the symmetric key (encrypted by $P_{n_i}^u$) to node n_i , gateway G_j, includes its arbitrary nonce, that is, N_{G_j} and N_{n_i} to the unicast message. Once this exchange is done, node n_i can ascertain that the message is recently initiated and is not a replay of old messages.

2.6 Survivable Secure Connectivity

In order to effectively present the connectivity in each cluster of the proposed infrastructure for WSNs, let a graph $G = (V, E)$ be defined to model the connectivity between a set of sensor nodes. Each sensor node is denoted by a vertex in $V, V = \{n_1, n_2, \ldots, n_{N_c}\}$, where N_c is the number of sensor nodes located within each cluster. For any random two nodes n_i and $n_{i'}$ in V, the edge $(n_i, n_{i'}) \in E$ exists if and only if the nodes are located within communication range of each other. Node degree is considered to be the number of edges connected to node. To further illustrate this, consider Figure 1, $\deg n_4 = 3$. Suppose the node n_i seeks to send information to node $n_{i'}$, let $P(n_i, n_{i'})$ represents the received power at $n_{i'}$. In this situation, the gateway G_j matches the SNR with the environment noise threshold, and if the result is above the noise threshold, then n_i can send a message to the $n_{i'}$. In this circumstance, these nodes have accomplished survivable connectivity and the edge $(n_i, n_{i'})$ exists. To obtain the $P(n_i, n_{i'})$ in each cluster, the following procedure is followed and completed.

1. The gateway broadcasts and initiate message
2. Each sensor node n_i sends message along with its ID_{n_i}
3. All sensor nodes keep log of the received signal strength
4. Sensor nodes are queried to report the recorded log to the gateway

To ensure secure connectivity, sensor nodes are expected to have previously established a symmetric secret common key $K_{n_i}^{n_{i'}}$ for each edge in E. Then, it can be ascertained that the proposed graph is securely connected and the gateway G_j will know the degree of each sensor node located within its cluster. It should be noted that the amount of symmetric keys whichis loaded from the gateway G_j to each sensor node is determined by $\deg n_i$.

2.7 Node Degree Analysis

The approach for establishing security for clustered WSNs is based on using public key cryptography (PKC). The required symmetric key for each sensor node is based on the node degree and routing algorithm. In this scheme, each sensor node has a secure path to the gateway across multiple hops. Hence, the degree of connectivity of each sensor node may vary. This routing algorithm depends on minimum neighborhood path, although some sensor nodes may have a higher neighborhood degree. However, it is important to show how many neighbors a sensor can possess.

The knowledge of the number of nodes in a certain area S in the surrounding of A is paramount. Since sensor nodes have arbitrary and even deployment, then it is not out of place to assume a Poisson distribution. Hence, the probability mass function can be expressed for the random deployment, as;

$$P(n \mid S) = Probability\ of\ n\ nodes\ is\ in\ area\ S \quad (6)$$

Considering node density and Poisson process, it can be written that

$$P(n \mid S) = \frac{(\rho S)^n}{n!} \cdot e^{-\rho S} = \frac{\left(\left({}^{N}/_{A}\right)S\right)^n}{n!} \cdot e^{-\left({}^{N}/_{A}\right)S} \quad (7)$$

Where $\rho = \left({}^{N}/_{A}\right)$. The average number of nodes within the radius r and area of $S = \pi r^2$ can be computed by

$$\bar{n} = \sum_{n=0}^{N} nP(n \mid S) = \rho \cdot S = \frac{N}{A}S = \frac{N}{A}\pi r^2 \quad (8)$$

To obtain the probability of having average number of sensor nodes within the environment of sensor node, the expression in Equation 9 can be applied.

$$Pr(n = \bar{n} \mid S) = \frac{(\rho \cdot S)^{\rho \cdot S}}{(\rho \cdot S)!} \cdot e^{-\rho S} \quad (9)$$

As $\rho \cdot S \gg 1$ considers the sterling's formula, it can be inferred that

$$Pr(n = \bar{n} \mid S) = \frac{1}{\sqrt{2\pi\rho \cdot S}} \quad (10)$$

It is pertinent to note that the density of sensor nodes after the clustering is the same since the deployment of sensor nodes is arbitrarily uniform. To compute the probability that each sensor node has at least n neighbors, the minimum node degree can be expressed as follows:

$$Pr(d \geq n) = (1 - \sum_{D=0}^{n-1} P(D \mid S))^N \quad (11)$$

To better illustrate this idea, assume that $N = 1000$ nodes are to be deployed arbitrarily in an area $A = 1000 \times 1000\ m^2$ and the range of transmission for each sensor node $r = 100\ m$. From Equation 8, the average number of surrounding nodes is $\bar{n} \approx 32$, and the probability of having this as surrounding degree is about 7.2% according to Equation 10. It should be noted that $\deg(n_i)$ and the number of symmetric keys that should be dynamically stored in each sensor node consequently are defined by the number of the neighboring nodes.

Again, as illustrated in Figure 1, the one-hop neighbors for the gateways G_1 and G_2 are $\{n_1, n_3, n_7\}$ and $\{n_8, n_{10}, n_{14}\}$, respectively. In order to set up secure connection between nodes in the routing path, the gateway G_1 will transmit secret key to the sensor node within its cluster by encrypting them with the public key of the given node. As illustrated in Figure 1, one-hop neighbors of sensor node n_{10} are $\{n_{11}, n_{12}, n_{13}\}$, thereafter, it receives $\{K_{n_{11}}^{n_{10}}, K_{n_{12}}^{n_{10}}, K_{n_{13}}^{n_{10}}\}$ symmetric keys encrypted with $P_{n_{10}}^{u}$. All sensor nodes within the network obtains the secret key distributed within their surrounding nodes similarly.

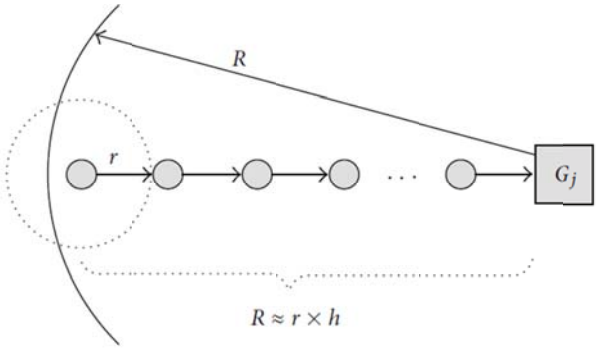

Figure 4: Approximating the cluster size from the number of hops and average node degree of each sensor node

2.8 Average Number of Sensor Nodes and Number of Hops in a Cluster

Since the sensor nodes are assumed to be evenly deployed on site, the following approximation is proposed for the average number of nodes per cluster and cluster size. Give that N_c denotes the number of the sensor nodes inside a cluster having the radius R. It is obvious that, N_c is according to the Poisson distribution similar to the node degree analysis presented in Equation 7. Then, N_c can be computed as

$$\overline{N_c} = \frac{N}{A}\pi R^2 \quad (12)$$

Where $\overline{N_c}$ represents the average number of sensor nodes within the cluster. By applying $R = h \times r$, Equation 12 can be rewritten as:

$$\overline{N_c} = \frac{N}{A}\pi h^2 r^2 \quad (13)$$

Where h denotes the maximum number of hops between a node and a gateway as depicted in Figure 4. By considering Equation 8, it can be inferred that

$$\overline{N_c} = \bar{n}h^2 \quad (14)$$

Then, the number of hops can be approximated as

$$h = \left\lfloor \sqrt{\frac{\overline{N_c}}{\bar{n}}} \right\rfloor \quad (15)$$

It is worthy to mention that in a real scenario with a fixed range of gateway, R, increasing the range of each sensor node, r, should be accompanied by reducing the quantity of hops to conserve energy and node lifetime. Hence, the average number of sensor nodes inside a cluster is constant.

2.9 Link Compromise Probability

The previously proposed schemes were built on the foundation of probabilistic key pre-distribution, and there is a known trade-off between the secure connectivity, resiliency against node capture, and memory storage.

Suppose x nodes are arbitrarily setup within a cluster. Then, resiliency could be defined in this context as the probability that the link between two fixed non-compromised nodes is not affected. The inverse of resiliency is coined which is also known as the fraction of the network that can be compromised. In multi-hop routing, it is basically obvious that choosing short multi-hop paths in place of long multi-hop paths is advantageous. This is due to the fact that the length of a multi-hop path (number of hops) increases as the probability of path compromise increases. Therefore, for the proposed scheme, it is necessary to compute the probability of the link between sensor node n_i and gateway G_j to be compromised without capturing them directly. Let us assume the following:

i. x_i: denotes the probability of node n_i to be compromised
ii. h: denotes the number of hops from a sensor node n_i to reach the gateway G_j

Hence, the probability that the given path which is compromised $P(l)$, assuming that the sensor node n_i and gateway G_j are not compromised, can be expressed as

$$\begin{aligned} &P(l) \\ &= Pr[\textit{link between sensor node } n_i \textit{ and the gateway } G_j \textit{ is compromised}] \\ &\quad 1 - Pr[\textit{no node in between is compromised}] \\ &\quad 1 - \prod_{i=1}^{h-1}(1 - x_i) \end{aligned} \quad (16)$$

After establishing the routing algorithm, the probability of node compromising directly or indirectly will be different since the number of sensor nodes in neighborhood is different. This compromise probability is based on the attacker model. Since our routing algorithm is based on minimum surrounding degree, the degree of each node is reduced to consequently decrease the indirect link compromise probability and have better resiliency against node capture attack.

2.10 Storage Saving Measurement

The memory storage requirement in sensor nodes and gateways are analyzed in this section. Considering the proposed network model, the number of gateways are far less than the number of sensor nodes $G \ll N$. Once the gateway is preloaded with $\left\{P_{G_j}^u, P_{G_j}^r, P_{n_i}^u\right\}$, then the memory storage requirement for each gateway can be computed as

$$M_G = (2 + N) \times B^u \quad (17)$$

Where, B^u denotes the key size for public cryptography.

Conversely, each sensor n_i is preloaded with $\left\{P_{n_i}^u, P_{n_i}^r, P_{G_j}^u\right\}$. During post deployment phase, each sensor node stores extra symmetric keys to communicate with their neighbors. This key can be represented as $\left\{K_{n_i}^{n_{i'}}\right\}$,

$$M_n = (G + 2) \times B^u + d_m \times B^k \quad (18)$$

Where B^k, represents the size of symmetric cryptography and d_m denotes the maximum neighborhood degree.

Since the gateways are tamper proof, the number of keys stored in each sensor node can be further reduced by incorporating the same pair of private and public keys for all the gateways, that is, P_G^r and P_G^u. Hence, the overall memory storage requirement for each sensor node can be expressed as

$$M_n = 3 \times B^u + d_m \times B^k \quad (19)$$

2.11 Communication and *Computation* Overheads

Intrinsically, randomized key pre-distribution approaches from previous literatures suffer from lack of structure because the key ring k is chosen randomly from a key pool. As a result, the communication complexity denoted as $\Theta(k)$, and increasing k yields a dramatic increase in communication overhead. The number of messages transferred in the network is a metric which relates to the power consumption and communication overhead. It is obvious that transmission is the most costly operation on a sensor node (for instance, the cost of transmitting one bit of data using MICA mote sensor node is approximately equivalent to processing 1000 CPU instructions) [12]. Hence, in this paper the communication overhead is defined as the sum of packets sent and received per cluster in the network. The average number of packets can be represented as the sum of the following.

i. Packet transmitted from G_j to n_i as message B in each cluster
ii. Packet transmitted by each sensor node towards the gateway within the cluster as message A
iii. Unicast encrypted messages (pairwise secrete keys) that each gateway sent to the nodes within its cluster $\left(K_{n_i}^{n_{i'}}\right)$

2.12 Cost of Secure Clustering and Pairwise Key Establishment

The number of encryptions and decryptions during secure clustering and pairwise key establishment is presented in

Table 1. Hence the cost of secure clustering C_{SC} can be computed as follows.

$$C_{SC} = G \times C_{ECDS_{P^r_{G_j}}} + N \times C_{ECDSV_{P^u_{G_j}}} + N \times C_{E_{P^u_{G_j}}}(\cdot) + N \times C_{D_{P^r_{G_j}}}(\cdot) \quad (20)$$

Where, $C_{ECDS_{P^r_{G_j}}}$ denotes the cost of generating an elliptical curve digital signature with the use of private key of gateway G_j, $C_{ECDSV_{P^u_{G_j}}}$ denotes the cost of verifying the signature using the public key of gateway G_j by sensor node n_i, $C_{E_{P^u_{G_j}}}(\cdot)$ denotes the cost of encryption using public key of gateway G_j by sensor node n_i, and $C_{D_{P^r_{G_j}}}(\cdot)$ denotes the cost of decryption using the private key of the gateway G_j performed by the gateway G_j.

Table 1: Number of encryption/decryption during secure clustering and pairwise key establishment. Note: elliptical curve digital signature is abbreviated as ECDS

Operation	No. of Computations
Secure Clustering	
ECDS generation and broadcast $G_j \rightarrow n_i$	G
ECDS verification by n_i	N
Encryption $E_{P^u_{G_j}}, n_i \rightarrow G_j$	N
Decryption $D_{P^r_{G_j}}(\cdot)$ by G_j	N
Pairwise key establishment	
ECDS and encryption by $E_{P^u_{n_i}}(\cdot), G_j \rightarrow n_i$	G
ECDS verification and decryption by $D_{P^r_{n_i}}(\cdot)$	N

3. Results and discussion

3.1 Performance Evaluation of Key Distribution

The proposed key distribution scheme is compared with other existing and commonly used distribution techniques. The result has proven that while providing similar probability of key sharing among nodes, the proposed key distribution scheme scheme significantly minimizes the storage requirements. The key pool size $\|K\|$ is a crucial parameter because in arbitrary key sharing schemes the amount of storage reserved for keys in each node is likely to be a preset constraint, which makes the size of the key ring $\|R\|$ a constant parameter. After R is set, then for larger values of $\|K\|$, the probability that two legitimate nodes will share a key is small. In addition, the probability that a randomly selected link is compromised when a node that is at neither end of the compromised link decreases by increasing the value of $\|K\|$.

In Figure 5, the range of key pool size is from 1,000 to 50,000 and key ring size is fixed to 100 for basic scheme proposed in [12]. For asymmetric pre-distribution AP scheme proposed in [12], sensors with high resources (H-sensor) keys are 500 and sensors with low resources (L-sensor) keys are 20. For the proposed scheme elliptical curve digital signature algorithm (ECDSA) in this research, the number of key chains (M) varies from 100 to 1,000, $S = 90$, and $r = 2$. Then, the number of key chains $(M) = 0.02$ times of the corresponding key pool size. Figure 5 also depicts that for the proposed scheme, the same probability of key distribution among nodes can be accomplished by just loading 2 generation keys in sensor node as compared to 100 keys in basic scheme, and 20 keys in AP scheme. For example, if there exist 10 H-sensors and 1000 L-sensors in an heterogeneous sensor network (HSN), where each L-sensor is preloaded with 2 generation keys and each H-sensor is preloaded with 100 generation keys, the total memory requirement for our proposed scheme in the unit of key length is 2 × 1000 + 100 × 10 = 3,000. However, in AP strategy, if each L-sensor is loaded with 10 keys and each H-sensor is loaded with 500 keys, the total memory requirement for storing these keys will be 500×10+ 1000×20 = 25,000, which is about 8 times larger than the proposed scheme. In addition, for a homogeneous sensor network with 1,000 L-sensors, where each L-sensor is preloaded with 100 keys, the memory requirements will result in 100 × 1000 = 100,000, which is 33 times larger than the proposed scheme.

Figure 6 illustrates that the probability of key distribution among nodes and gateways increases by a minute increase in the number of preloaded generation keys in L-sensors. For example, if preloaded keys are increased from 2 to 5, the key distribution probability increases from 0.5 to 0.8 approximately, for 400 key chains.

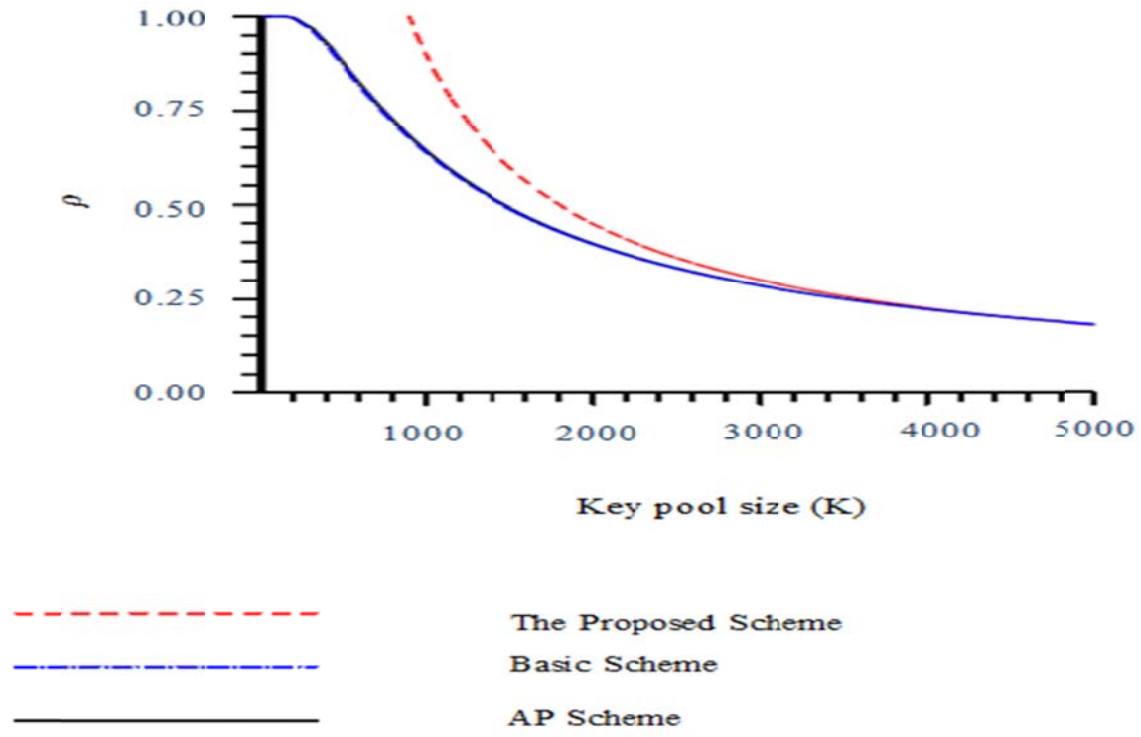

Figure 5: The probability of key sharing among sensor nodes

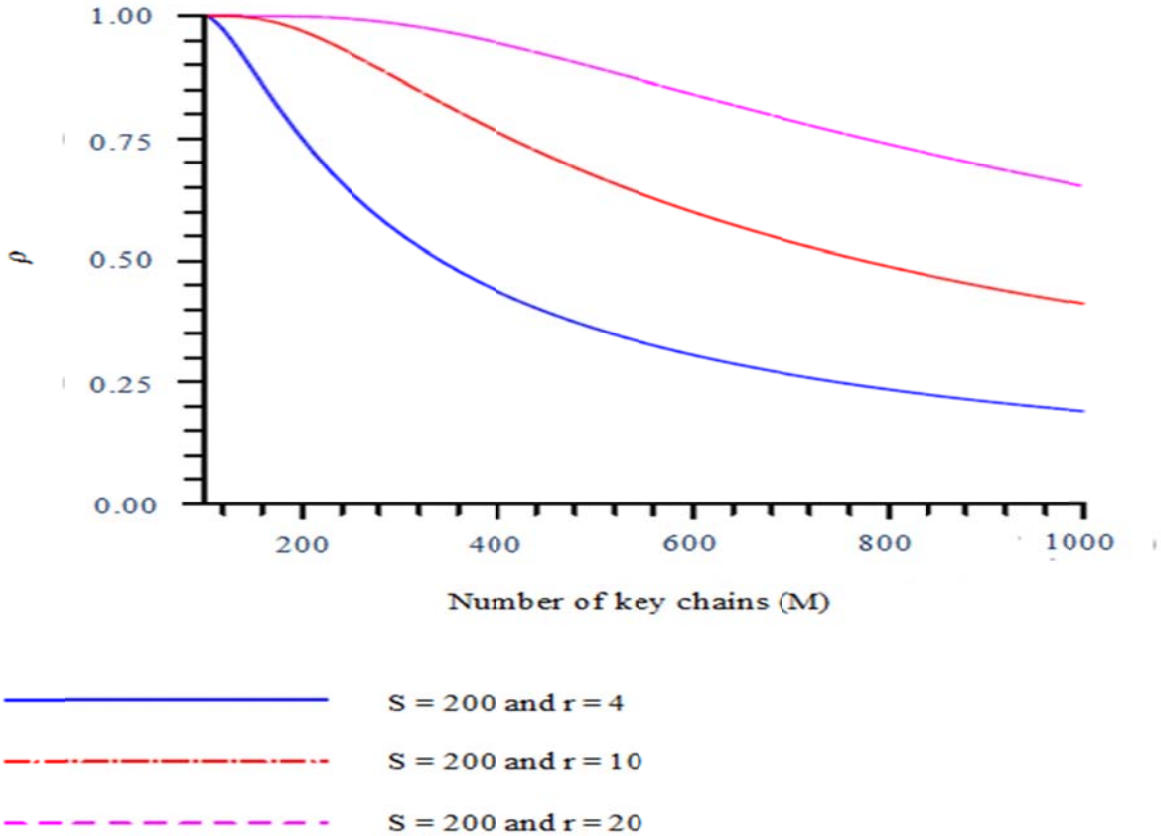

Figure 6: The probability of key sharing between sensor nodes and gateway

3.2 Performance Analysis of Link Compromise Probability

Recall that resiliency was defined as the probability that the link between two fixed (in a randomly distributed scenario) non-compromised nodes is not affected. Once the routing algorithm is established, the probability of node compromise directly or indirectly will be different since the number of sensor nodes within the neighborhood is different. In Figure 7, the impact of increasing, number of hops on link compromise probability is shown in terms of

node compromise probability x_i. Since the proposed routing algorithm is based on minimum neighborhood degree, effort is made to reduce the degree of each node to decrease the indirect link compromise probability and have better resiliency against node capture attack. For simulation purpose, it is assumed that a network with $N = 1000$ sensor nodes is randomly and uniformly installed in an area of $A = 1000 \times 1000\ m^2$. The number of gateways is selected as $G = 10$ to cover a reasonable area of sensor nodes. A variable transmission range is adopted as $r = 25\ m$ to $r = 100\ m$ to obtain different average node degree $\bar{n}$ ranging from 2 to 32. The maximum range is set to $R = 200\ m$. This simulation was performed using QualNet simulator. QualNet simulator is a scalable wireless sensor network simulator.

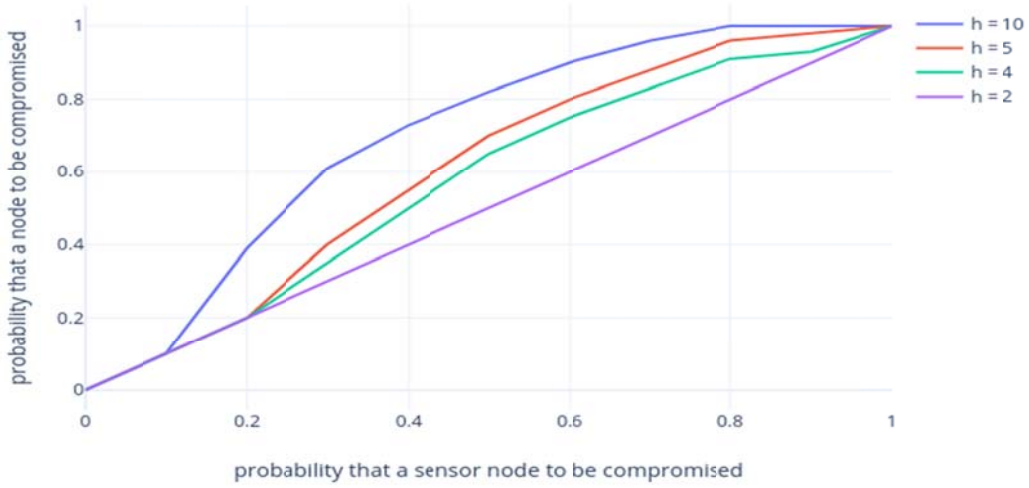

Figure 7: The effect of number of hops on link compromise

Figure 8: Number of neighbor nodes involved in the routing algorithm towards the gateway with $N = 1000$, $G = 10$, and $r = 100\ m$

By the simulation result, the number of neighbor nodes which are involved in the routing algorithm and are communicating securely (using allocated symmetric keys) is shown. The secure neighborhood degree is plotted for each sensor node for the proposed network model in Figure 8. About 300 nodes communicates with just two sensor nodes and about 25 sensor nodes securely communicate with 7 other neighbor nodes. The simulations were executed three times, and the results obtained are almost the same. Hence, the highest number of symmetric keys which are required to be dynamically loaded to the sensor nodes is always less than the average number of nodes $\bar{n}$ for the proposed scheme.

3.3 Performance Analysis of Sensor Nodes Range Variation Effect on Number of Hops

The number of sensor nodes in a cluster is computed using Equation 12 to Equation 14. Recall it was mentioned that in a real-time case with a fixed range of gateway R, increasing the range if each sensor node r, must be followed by decreasing the number of hops for energy saving purpose.

The range of sensor nodes is varied from 25 m to 100 m to obtain the relevant number of hops as illustrated in Table 2.

Table 2: Analytical number of hops with various sensor node transmission ranges for a fixed gateway range $R = 200$.

r	$\bar{n}$	$\overline{N_c}$	h
25	2	128	8
50	8	128	4
75	18	128	3
100	32	128	2

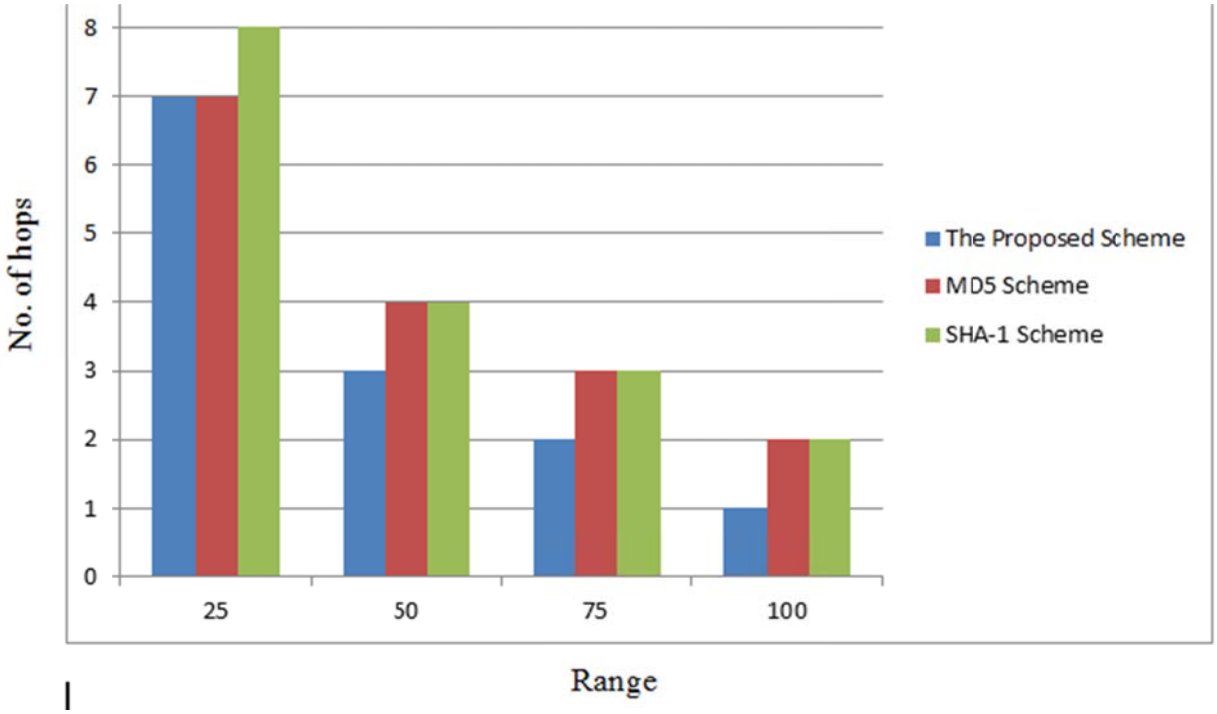

Figure 9: Analytical number of hops with various sensor node transmission ranges for a fixed gateway range $R = 200$.

Figure 9 is plotted based on the data in Table 2. The proposed scheme which uses approximation based on the Poisson distribution is compared to the MD5 scheme which is proposed in [14] and the SHA-1 scheme in [8]. It is shown that for short range ($\leq 25\ m$) MD5 scheme and the proposed scheme has the same number of hops which the SHA-1 scheme has higher. However, for longer ranges ($\geq 50\ m$), the proposed scheme proves its excellence. In this case, the proposed scheme requires lower number of hops, hence, minimizes the probability of compromise and also saves sensor nodes energy, in addition.

3.4 Memory Requirement Evaluation

The proposed scheme (ECDSA) in this research requires less memory space as compared to the probabilistic scheme proposed in [14], where those schemes requires $m \times B^k$ bits. To demonstrate this, the proposed scheme (163-bits) will be used between sensor nodes and gateway and the SKIPJACK (83-bitss) cryptography is used in communication between sensor nodes and its neighbors. The results obtained from the probabilistic scheme will be compared. Considering Equation 18 and Equation 19, the worst case memory requirement for each sensor node is $M_n = (3) \times 163 + 7 \times (83) = 1{,}070\ bits$. As shown in Figure 8, the maximum node degree in the proposed scheme is 7. In the probabilistic scheme, the storage requirement is $(54) \times 83 = 4482\ bits$ for balanced scheme and $30 \times 83 = 2490\ bits$ for unbalanced scheme with connectivity of 67%. Hence, the proposed approach saves about 57% of memory storage compared to the probabilistic scheme.

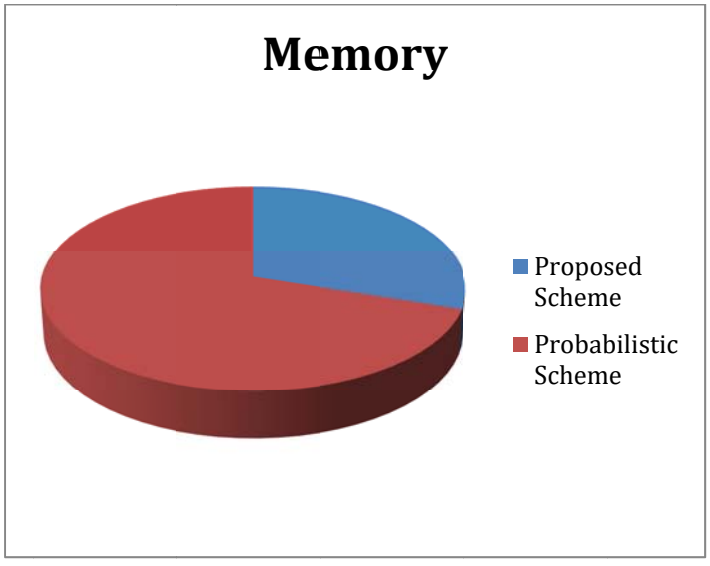

Figure 10: Memory Requirements for sensor nodes

This analysis is depicted in Figure 10. It is worth mentioning that the proposed scheme is deterministic and

totally connected. From Equation 17, it can be deduced that the number of keys stored in each gateway is 1002 keys.

4. Conclusion

An approach for securing symmetric key used in clustered Heterogeneous Wireless Sensor Networks (HWSNs) using elliptical curve digital signature algorithm (ECDSA) is presented. The network model along with explanation regarding secure clustering and symmetric key establishment in the HWSNs are presented along with elaboration on how security is established in the initial phase of bootstrapping and clustering of these networks. Relevant mathematical models pertaining to the proposed ECDSA scheme are presented and then the performance of the ECDSA key distribution scheme is compared with other existing and commonly used distribution techniques. The results show that while providing similar probability of key sharing among nodes, the ECDSA scheme significantly minimizes the storage requirements. It also minimizes the probability of compromise and also saves sensor nodes energy.

References

[1] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2314–2341, 2017.

[2] L. Oliveira, H. Wong, A. Loureiro, and R. Dahab, "On the design of secure protocols for hierarchical sensor networks,"
International Journal of Security and Networks, vol. 2, no. 3, pp. 216–227, 2017.

[3] *IEEE Std.802.15.4 for Information Technology—Telecommunication and Information Exchage between Systems—Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical*
Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks, IEEE, 2018.

[4] V.Westmark, "A definition for information system survivability," in *Proceeding of the 37th Hawaii Internal Conference on System Sciences (HICSS '04)*, pp. 2086–2096, IEEE press, 2020.

[5] K. Akkaya andM. F. Younis, "Energy andQoS aware routing in wireless sensor networks," *Cluster Computing*, vol. 8, no. 2-3, pp. 179–188, 2019.

[6] C. P. Low, C. Fang, J. M. Ng, and Y. H. Ang, "Efficient loadbalanced clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 31, no. 4, pp. 750–759, 2018.

[7] M. F. Younis, K. Ghumman, and M. Eltoweissy, "Locationaware combinatorial key management scheme for clustered sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 865–882, 2018.

[8] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions onMobile Computing*, vol. 3, no. 4, pp. 366–379, 2018.

[9] Q. Tian and E. J. Coyle, "Optimal distributed detection in clustered wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3892–3904, 2017.

[10] J. Deng and Y. S. Han, "Multipath key establishment for wireless sensor networks using just-enough redundancy transmission," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 3, Article ID 4378397, pp. 177–190, 2018.

[11] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02)*, pp. 41–47, ACM, November 2021.

[12] K. Lu, Y. Qian, M. Guizani, and H.-H. Chen, "A framework for a distributed key management scheme in heterogeneous wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 2, pp. 639–647, 2018.

[13] W. Zhang, M. Tran, S. Zhu, and G. Cao, "A random perturbationbased scheme for pairwise key establishment in sensor networks," in *Proceedings of the 8th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing,*
(MobiHoc '07), E. Kranakis, E. M. Belding, and E. Modiano, Eds., pp. 90–99, ACM, 2020.

[14] Azarderskhsh, R., & Reyhani-Masoleh, A. (2011). Secure clustering and symmetric key establishment in heterogeneous wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, *2011*, 1-12.

[15] Shi, Q., Zhang, N., Merabti, M., & Kifayat, K. (2013). Resource-efficient authentic key establishment in heterogeneous wireless sensor networks. *Journal of parallel and distributed computing*, *73*(2), 235-249.

[16] Kim, D., Kim, D., & An, S. (2016). Communication pattern based key establishment scheme in heterogeneous wireless sensor networks. *KSII Transactions on Internet and Information Systems (TIIS)*, *10*(3), 1249-1272.

[17] Zhou, R., & Yang, H. (2011, August). A hybrid key management scheme for Heterogeneous wireless sensor networks based on ECC and trivariate symmetric polynomial. In *2011 International Conference on Uncertainty Reasoning and Knowledge Engineering* (Vol. 1, pp. 251-255). IEEE.

[18] Kumar, K. A., Krishna, A. V., & Chatrapati, K. S. (2017). New secure routing protocol with elliptic curve cryptography for military heterogeneous wireless sensor networks. *Journal of Information and Optimization Sciences*, *38*(2), 341-365.

[19] Boujelben, M., Cheikhrouhou, O., Youssef, H., & Abid, M. (2009, June). A pairing identity based key management protocol for heterogeneous wireless sensor networks. In *2009 International Conference on Network and Service Security* (pp. 1-5). IEEE.