American Journal of Engineering Research (AJER)
e-ISSN : 2320-0847 p-ISSN : 2320-0936
Volume-03, Issue-02, pp-191-203
www.ajer.org

Research Paper Open Access

A Secure Hierarchical Protocol for Wireless Sensor Networks

Poulami Dutta
(Department of Computer Science and Engineering, Techno India Salt Lake, Kolkata, India,

Abstract: - The literature for considering routing protocols in wireless sensor networks (WSNs) is very broad. However, security of these routing protocols has fallen beyond the scope so far. Routing is a fundamental functionality in wireless networks, thus hostile interferences aiming to disrupt and degrade the routing service have a serious impact on the overall operation of the entire network. Gathering sensed information in an energy efficient manner is also critical for increasing the lifetime of the network. We propose a formal framework for the security analysis of a clustering-based hierarchical routing protocol, S-PEGASIS, for wireless sensor networks. Our approach is based on the well-known simulation paradigm that has been used to enhance the security of the protocol. Our main contribution is the application of the simulation approach to incorporate security into the model in order to make it more energy efficient as compared to the others in its class. We further point out that our protocol is scalable and its transmission time complexity is logarithmic in nature. We also point out that security can be further extended by providing link layer and physical layer security mechanisms so that trusted nodes on the chain (S-PEGASIS) can never be hacked.

Keywords: - *Clustering, Energy Efficiency, Hierarchical, Routing Protocols, Security, Wireless Sensor Networks*

I. INTRODUCTION

Sensor networks are dense wireless networks of small, low-cost sensors, which collect and disseminate environmental data. Wireless Sensor Networks (WSNs) consist of numerous tiny sensors deployed at high density in regions requiring surveillance and monitoring. They are deployed at a cost much lower than the traditional wired sensor system. A large number of sensors deployed will enable for accurate measurements. A sensor node consists of one or more sensing elements (motion, temperature, pressure, etc.), a battery, and low power radio trans-receiver, microprocessor and limited memory, mobilizer (optional), a position finding system. An important aspect of such networks is that the nodes are unattended, have limited energy and the network topology is unknown. The emerging field of wireless sensor networks combines sensing, computation, and communication into a single tiny device. The power of wireless sensor networks lies in the ability to deploy large numbers of tiny nodes that assemble and configure themselves.

Recent advances in micro-electro-mechanical systems and low power and highly integrated digital electronics have led to the development of micro-sensors. Such sensors are generally equipped with data processing and communication capabilities. The sensing circuitry measures ambient conditions related to the environment surrounding the sensor and transform them into an electric signal. Processing such a signal reveals some properties about objects located and/or events happening in the vicinity of the sensor. The sensor sends such collected data, usually via radio transmitter, to a command center (sink) either directly or through a data concentration center (a gateway). The decrease in the size and cost of sensors, resulting from such technological advances, has fuelled interest in the possible use of large set of disposable unattended sensors. Such interest has motivated intensive research in the past few years addressing the potential of collaboration among sensors in data gathering and processing and the coordination and management of the sensing activity and data flow to the sink. A natural architecture for such collaborative distributed sensors is a network with wireless links that can be formed among the sensors in an ad hoc manner.

Routing protocol is a set of rules defining the way a router finds the way that packets containing information have to follow to reach the intended destination. In this section, we discuss some of the routing protocols to be used in designing the WSN. In general, routing protocols can be considered to belong to one of the following models:

i) Data-centric protocols,
ii) Hierarchical protocols, and
iii) Location-based protocols.

In data-centric routing, the sink sends queries to certain regions and waits for data from the sensors located in the selected regions. Since data is being requested through queries, attribute-based naming is necessary to specify the properties of data. SPIN is the first data-centric protocol, which considers data negotiation between nodes in order to eliminate redundant data and save energy. Later, Directed Diffusion was developed and it has now become a breakthrough in data-centric routing. The main aim of hierarchical routing is to efficiently maintain the energy consumption of sensor nodes by involving them in multi-hop communication within a particular cluster and by performing data aggregation and fusion in order to decrease the number of transmitted messages to the sink. Cluster formation is typically based on the energy reserve of sensors and sensor's proximity to the cluster head. LEACH is one of the first hierarchical routing approaches for sensors networks along with a few others like PEGASIS, TEEN, and APTEEN. Most of the routing protocols for sensor networks require location information for sensor nodes. In most cases location information is needed in order to calculate the distance between two particular nodes so that energy consumption can be estimated. Since, there is no addressing scheme for sensor networks like IP-addresses and since they are spatially deployed on a region, location information can be utilized in routing data in an energy efficient way. For instance, if the region to be sensed is known, using the location of sensors, the query can be diffused only to that particular region which will eliminate the number of transmission significantly. An example of location based protocol is GEAR.

WSNs are prone to many types of attacks because they use wireless communication which is very insecure due to high probability of eavesdropping by attackers. Secondly, it is not possible to employ stringent security measures due to highly constrained resources. These attacks may occur at physical layer, data link layer or the network layer during communication between nodes. We are trying to develop a routing protocol which can detect and prevent such attacks. We will, however, concentrate on network layer attacks only. This model will be a hierarchical one. The advantage of using hierarchical protocol is that they are much more energy efficient as compared to other protocols and hence enhance lifetime of the network. Secondly, they are prone to smaller number of attacks as compared to other protocols due to a fixed path of data delivery and small number of hops each packet has to go. Thirdly, they provide better scalability due to less number of node-to-base station communications and corresponding network congestion.

Akyildiz et al. has reviewed the factors that influence the design of WSNs and has also surveyed the underlying architecture needed to execute the protocols developed in each layer of the protocol stack in. The design of WSNs is motivated by factors viz. fault tolerance, scalability, production costs, operating environment, underlying topology, transmission media and power consumption. The communication architecture has been researched upon. The protocol stack consisting of the 5 layers, application, transport, network, data link and physical, task management plane, mobility management plane and power management plane. It integrates functions like node coordination to accomplish the sensing task and lower the overall power consumption [1]. The power management plane helps in proper utilization of power by the nodes, which turns off its receiver after receiving a message from one of its neighbors. If the power level of the node is low, a message is broadcast to the neighboring nodes that the node in question cannot participate in routing. The balance power is reserved for sensing. The mobility management plane detects the trajectory of the nodes, registers its neighbors in order to balance power usage. Finally the task management plane schedules the sensing tasks. Some nodes are better performers as compared to the others and this totally depends on their power level. The 3 planes work together in a power efficient way to route data packets to the BS and share resources among themselves. The work examines some protocols in the different layers of the protocol stack. The sensor management protocol (SMP), task assignment and data advertisement protocol (TADAP), sensor query and data dissemination protocol (SQDDP) work in the application layer of the protocol stack. All these protocols are open research issues as explored by the authors in terms of time synchronization, network configuration, mobility of nodes, key distribution, security in data communication etc. The development of transport layer protocols remains a challenge, especially because of hardware constraints like limited power and memory, as a result of which the nodes cannot store large amounts of data like a server. Research to enable communication in this layer needs to be explored upon. It also provides insight into current routing protocols in the network layer like small minimum energy communication network (SMECN), sensor protocol for information via negotiation (SPIN), low-energy adaptive clustering hierarchy (LEACH), sequential assignment routing (SAR) and directed diffusion (DD). In a nutshell, these protocols proposed need to be improved in order to address topology changes, make the network more scalable, and allow easy communication between the network and the external networks i.e. allow internetworking. MAC protocols cannot be adopted into the WSN scenario due to its resource constraints and therefore they should be modified accordingly to suit the need of current state-of-art applications. Focus needs to be given on providing cross-layer security in the protocol stack, because careful security design in the lower layers can effectively prevent attacks at the application layer. While all the existing solutions can secure the WSN to a certain extent, there is no one solution that can satisfy the security goals completely and guarantee the integrity, authenticity and availability of messages in the presence of adversaries.

The authors in Akkaya et al. have categorized the routing protocols in WSNs on the basis of their modes of operations and have also presented a comparative study on them [2]. Routing techniques are required to send data and communicate between the BS (Base Station) and the sensor nodes. These protocols can be classified as proactive, reactive and hybrid depending upon their modus operandi and their target application. In the proactive category, the nodes switch on their sensors and transmitters, sense the environment and route data packets along pre-defined routes. Example: LEACH (Low Energy Adaptive Clustering Hierarchy). In the 2nd category, nodes react to sudden changes in the environment beyond a pre-determined threshold. Example: TEEN (Threshold Sensitive Energy Efficient Sensor Network). Hybrid protocol incorporates both proactive and reactive concepts. It first computes all routes and then improves the routes at the time of routing. Example: APTEEN (Adaptive Periodic TEEN). Routing protocols can be further categorized as direct, flat and clustering as per the participation style of the sensor nodes. In case of the 1st category, nodes can communicate directly with the BS. When the network grows larger in size, the nodes drain out faster. Example: SPIN. In the case of flat protocols if any node needs to transmit data, it first searches a valid route to the BS and then transmits data. Nodes around the base station may drain out of their energy quickly. Example: Rumor Routing. In the 3rd category, nodes are divided into clusters and each cluster has a CH (cluster-head) and these CHs communicate with the BS. Example: TEEN. Furthermore, depending on the network structure, protocols can be classified as hierarchical, data centric and location based. Hierarchical routing is used for increasing the energy efficiency of the WSNs. The high energy nodes are used for processing and transmitting and the low energy ones for sensing the AOI (Area of Interest). Example: LEACH, TEEN, APTEEN. Data-centric protocols are query-based and depend upon naming the data to query a certain AOI and thereby reduce redundant transmissions. Example: SPIN. Location-based routing protocols need location information of the sensor nodes, which can be obtained either manually or by installing a GPS (Global-Positioning System) in the node. Example: GEAR (Geographic and Energy-Aware Routing). This work has compared the protocols on the basis of parameters like mobility, power management, network lifetime, scalability, resource awareness, data aggregation, query-based and nature of routing. The survey sums up the fact that mobility is limited for DD and GEAR, whereas LEACH, TEEN, APTEEN and PEGASIS use the concept of a fixed BS. Power management is limited for SPIN, DD and GEAR and maximized for LEACH, TEEN, APTEEN and PEGASIS. Network lifetime is maximized for LEACH, TEEN, APTEEN, PEGASIS and RR. Scalability is good for LEACH, TEEN, APTEEN, PEGASIS and RR. All protocols surveyed are aware in terms if resource usage. Only SPIN, DD AND RR are query-based in nature and can therefore exploit this feature to eliminate redundancies while data routing. Routing technique used in SPIN and DD are multi-path in nature and hence these protocols are efficient in terms of energy usage and reliable in terms of data delivery to the destination, though the latter one is not quite applicable for SPIN. This work has done a survey on routing in WSNs and presented 8 different types of protocols in this domain. Future work can concentrate on modifying any of these protocols to suit a specific application in WSN by integrating wireless networks with wired ones, addressing issues of QoS in case of real-time applications, introducing node mobility to handle topology changes in case of critical applications like battle-fields, and also having efficient naming schemes for data-centric protocols.

Taxonomy of routing protocols in WSNs has been developed by Karaki et al. This taxonomy broadly classifies routing protocols according to the network structure and protocol operation. Protocols that fall into the former category are flat, hierarchical and location-based. Protocols that fall into the latter category are negotiation-based, multi-path based, query-based, QOS-based and coherent-based routing [3]. Apart from the traditional routing protocols, this paper provides an insight into protocols based on their operation. For multi-path routing, which was aimed at enhancing the reliability of the network, and hence useful for delivering data in vulnerable environments would increase traffic significantly. A trade-off using a redundancy function eliminating duplicates and conserving energy for the nodes thereby increasing their life time could be incorporated into this category of routing algorithms. This work also motivates us to use braided paths whose costs are comparable with those of the primary paths because of their proximity. In case of query-based routing the BS propagates a query to the other nodes in the network, which respond with matching data. Data aggregation is performed to lower the energy consumption. DD and RR belong to this category. The idea behind negotiation-based routing is to suppress duplicate information and prevent redundant data from being sent to the next sensor or BS by transmitting a series of negotiation messages before the actual data transmission begins. Eg: SPIN. In QoS routing, the network balances energy consumption with data quality to satisfy metrics like delay, energy, bandwidth etc. Eg: SAR. In coherent and non-coherent routing, various data processing techniques are used and sensor nodes cooperate with each other in processing data flooded across the WSN. In the former, nodes forward data to aggregators, which could be CHs after minimum processing, whereas in the latter, nodes locally process raw data before forwarding to others for further processing. Tiered architectures to maximize network lifetime is a hot area of research along with node deployment in an unpredictable environment to attain adaptive localization and coverage. Time synchronization and self-configuration are other possible future research domains for routing protocols.

While many surveys in existing literature address the issue of security in wireless networks, none focus specifically on the security issue in WSNs. The work in [4] analyzes the threats and security requirements and classifies the attacks as insider vs. outsider, active vs. passive and mote class vs. laptop class. It evaluates attacks in the various layers of the protocol stack and also suggests defense schemes against them. Selecting the most appropriate cryptographic method is vital in WSNs because all security services are ensured by cryptography. Cryptographic methods used in WSNs should meet the constraints of sensor nodes and be evaluated by code size, data size, processing time, and power consumption. These issues have also been addressed in [4]. It highlights some of the open research issues in WSNs like the application of private key operations to sensor nodes, design of efficient and flexible key distribution schemes based on symmetric key cryptography, schemes to ensure security of base stations, new schemes with higher scalability and efficiency for the authenticated broadcast protocols etc. New data aggregation protocols need to be developed to address higher scalability and higher reliability against aggregator and sensor node cheating. Also the proposed protocols on IDS (Intrusion Detection System) address the basic needs on how to filter fraudulent data from the system, can be improved to address the issue of scalability.

The authors in Pal et al. [5] have discussed the architecture of the wireless sensor network. Further, they have categorized the routing protocols according to some factors and summarize on their mode of operation. At the end, they put up a comparative study on these protocols. Routing in sensor networks is a new area of research. Since sensor networks are designed for specific applications, designing efficient routing protocols for sensor networks is very important. Since the sensor networks are application specific, we can’t say whether a particular protocol is better than other. We can only compare these protocols with respect to some parameters only as has been done here. Future work can concentrate on modifying any of these protocols to suit a specific application in WSN addressing issues of QoS in case of real-time applications, introducing node mobility to handle topology changes in case of critical applications like battle-fields, and also having efficient naming schemes for data-centric protocols.

Karlof et al. has analyzed the issue of security for routing in WSNs by proposing threat models and security goals, by introducing 2 new types of attacks, viz. sinkhole attacks and HELLO FLOOD attacks and finally crippling attacks and have also suggested counter measures against all of them. A summary of attacks against the protocols have been presented in [6]. Tiny OS beaconing is subjected to selective forwarding, sinkhole, and wormhole and HELLO flood attack. DD is also susceptible to all the above mentioned types of attacks. GEAR suffers from selective forwarding and Sybil attacks, clustering protocols like LEACH, TEEN and PEGASIS suffer from selective forwarding and HELLO flood attacks, and RR is threatened by Sybil and HELLO flood attacks. This work gives a clear idea of security issues in routing in WSNs and aims at ensuring that a secure routing protocol should be able to guarantee the security primitives like integrity, authenticity, and availability of messages in the presence of adversaries. A few common attacks like, spoofing, alteration, replaying of routing information, selective forwarding, sinkhole, Sybil, wormholes, HELLO flood have been extensively studied. By spoofing, altering and replaying routing information, adversaries (ADVs) are able to create routing loops to either attract or repel network traffic to either lure nodes into sending data packets or mislead them from reaching their targets. In case of selective forwarding attacks, ADVs suppress traffic from reaching their destinations by dropping them. Sinkhole attacks work by making a compromised node look attractive in order to lure traffic towards it. In WSNs, all nodes have the same ultimate destination, i.e. the BS, so it becomes easier for the ADV to lure traffic from its neighbors to be route through it and finally provides a single-route to the BS. A malicious node impersonates itself to other nodes in the network and thereby reduces the efficacy of the WSN from being fault-tolerant. This attack is harmful for geographic routing protocols. In case of wormhole attacks, ADV channels packets received in one part of the network to the other and disrupts routing by creating a wormhole. A wormhole is an illusion used to convince two distant nodes that they are within the radio range of one another and thereby force them to communicate by relaying data packets to each other. In case of HELLO flood attacks, nodes broadcast HELLO packets to their neighbors. ADV takes advantage of this and advertises itself in the radio range thereby causing nodes to attempt communication via this route and in the process wreaks havoc in the network. How these attacks affect the functionalities of the protocols by disembarking traffic to cause mayhem has also been stated as a part of the survey. Though countermeasures have also been proposed to combat attacks in these routing protocols, by using link layer encryption, authentication techniques and or cryptography, but they are not enough to defend against all categories of attacks. So this work gives opportunities to design a secure routing protocol which can suffice the demands of current state-of-art applications.

An optimal chain-based solution that is an improvisation over LEACH is PEGASIS (Power Efficient GAthering in Sensor Information Systems) as established by Lindsey et al. The idea is to form a chain of nodes where each node sends and receives from the next neighbour node on the chain. The concept of data fusion is used to combine data from different nodes into one data packet that only one designated node transmits to the BS. Nodes take turns in transmitting to the BS in order to optimize the energy consumption and thus increase the lifetime of the network. The leader that takes turns in transmitting to the BS which is placed at random positions on the chain [7]. This is to ensure that the WSN is robust to failures. The chain formation is done using a greedy algorithm to optimize the utilization of energy and bandwidth and also to outperform the other protocols in this genre.

In [8], a formal model of security for routing protocols in WSNs has been proposed. This new model differs from its previous versions in the sense that it adapts itself to suit the characteristics of the WSN. A major breakthrough of this work is the definition of the output of the model proposed as a function of the routing state of the honest nodes. This can put all categories of routing protocols on the same platform. As a case study this work has considered the Tiny OS beaconing protocol to illustrate how an attack can be formulated using the formal model and also establishes its veracity using the simulation paradigm. How the statistical model can be used to counter the effect of attackers in the other categories of routing protocols established in literature have not been discussed.

Another approach presented in [9], proposes a formal framework for the security analysis of on-demand source routing protocols for wireless ad hoc networks. It gives a formal definition of routing security in terms of indistinguishability of the two models from the point of view of honest parties. The approach is demonstrated by analyzing two secure ad hoc routing protocols, SRP (Secure Routing Protocol) and Ariadne. It proposes a routing protocol that can prove to be secure in this model. A particularly interesting direction for future work is trying to automate the analysis of ad hoc routing protocols using this simulation paradigm.

We have designed a new protocol called secure PEGASIS (or S-PEGASIS) which incorporates security in the hierarchical framework. The model designed has been simulated using NS2 and results obtained compared with existing ones to see how energy efficient and secure our design is.

II. OVERVIEW OF CLASSICAL PEGASIS AND HIERARCHICAL PEGASIS

It is the drawback/s of the first protocol of its kind in this genre of hierarchical routing, eg: LEACH (Low Energy Adaptive Clustering Hierarchy) like single-hop routing that makes it unsuitable for deployment over large areas, dynamic topology adjustment and CH assignment that involves significant overhead that curbs some of the energy gain and complex calculations in each round of data transfer, which requires a perfect random number generator, that a new protocol called PEGASIS (Power Efficient Energy Gathering in Sensor Information Systems) was developed. It is an improvement over LEACH protocol. In LEACH, 5% of nodes acted as cluster heads. Here, only one node communicates with the BS. Each node transmits only to its local neighbor in the data fusion phase.

The key idea in PEGASIS [7] is formation of chains where each node receives from and transmits to a close neighbor. Data travels from node to node, getting fused at each step with the host node's data and is eventually transmitted to the BS by a designated node called leader for that round. Nodes take turns in transmitting to the BS (being the leader) as it is the most energy consuming process (assuming that the BS is far away). This reduces the average energy consumption of each node per round. To have balanced energy dissipation in nodes, the remaining energy level of nodes is also considered as a parameter in addition to transmission-energy (cost) for determination of leader in each round or close neighbor. Long chains in PEGASIS may cause delay for nodes far from leader. To curb this, the concept of hierarchical PEGASIS has been introduced where multiple data transmissions may occur in parallel.

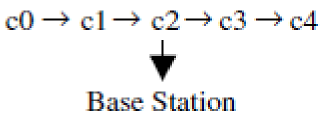

Figure 1: PEGASIS

The first step is the formation of chain of nodes using a greedy approach that works well if the nodes are randomly distributed. If there are N nodes in the network (we assume they are numbered), they become leaders by taking turns. A node may not be allowed to become a leader if its remaining energy level is low or closest neighbor on chain has a distance above a specified threshold. The chain may be constructed by the nodes themselves using greedy algorithm starting from some node or by the BS (and then broadcasted). Each time a node dies, a new chain has to be constructed bypassing it.

We illustrate the data transmission mechanism using an example network with a few nodes. As shown in figure, node c0 passes its data to node c1. Node c1 aggregates node c0's data with its own and then transmits to the leader. After node c2 passes the token to node c4, node c4 transmits its data to node c3. Node c3 aggregates node c4's data with its own and then transmits to the leader. Node c2 waits to receive data from both neighbors and then aggregates its data with its neighbors' data. Finally, node c2 transmits one message to the base station.

However, PEGASIS introduces excessive delay for distant node on the chain. In addition the single leader can become a bottleneck. Hierarchical-PEGASIS is an extension to PEGASIS, which aims at decreasing the delay incurred for packets during transmission to the base station and proposes a solution to the data gathering problem by considering energy delay metric. In order to reduce the delay in PEGASIS, simultaneous transmissions of data messages are pursued. The chain-based protocol with CDMA capable nodes, constructs a chain of nodes, that forms a tree like hierarchy, and each selected node in a particular level transmits data to the node in the upper level of the hierarchy. This method ensures data transmitting in parallel and reduces the delay significantly. Since the tree is balanced, the delay will be in O ($\log_2 N$) where N is the number of nodes.

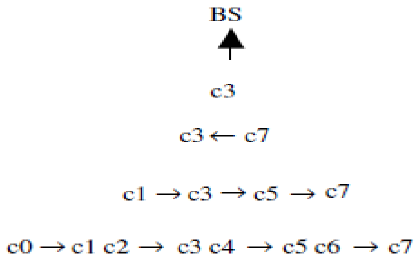

Figure 2: HIERARCHICAL PEGASIS

For example, in the above diagram, node c3 is the designated leader for round 3. Since, node c3 is in position 3 (counting from 0) on the chain, all nodes in an even position will send to their right neighbor. Nodes that are receiving at each level rise to next level in the hierarchy. Now at the next level, node c3 is still in an odd position (1). Again all nodes in an even position will aggregate its data with its received data and send to their right. At the third level, node c3 is not in an odd position, so node c7 will aggregate its data and transmit to c3. Finally, node c3 will combine its current data with that received from c7 and transmit the message to the sink. The non-CDMA based approach creates a three-level hierarchy of the nodes and interference effects are reduced by carefully scheduling simultaneous transmissions. Such chain-based protocol has been shown to perform better than the regular PEGASIS scheme by a factor of about 60.

PEGASIS improves on LEACH by saving energy in several stages [1], [2], [7]. First, in the local gathering, the distances that most of the nodes transmit are much less compared to transmitting to a cluster-head in LEACH. Second, the amount of data for the leader to receive is at most two messages instead of 20 (20 nodes per cluster in LEACH for a 100-node network). Finally, only one node transmits to the BS in each round of communication. However, the PEGASIS still requires dynamic topology adjustment since sensor's energy is not tracked. For example, every sensor needs to be aware of the status of its neighbor so that it knows where to route that data. Such topology adjustment can introduce significant overhead especially for highly utilized networks.

Furthermore, any hierarchical routing protocol is prone to an intrusion attack namely HELLO FLOOD attack which in turn may lead to selective forwarding attack. Routing protocols often use HELLO packet broadcasts by nodes during establishment of clusters. If a node A receives such a packet from another node B, A considers B to be within its normal radio range. Clustering is then done by picking the nearest neighbor in case of PEGASIS chain formation or choosing the CH from which the strongest signal is received in case of LEACH. But if a strong signal emitting adversary is present which broadcasts HELLO packets, the other nodes may mistake it for a legitimate neighbor and add it in the chain/choose it as cluster head. This node can spoof and misinterpret data received from its neighbor nodes/cluster members. It may forward spoofed packets along the route leading to selective forwarding. When this node takes turn and becomes the leader in the chain in case of PEGASIS, it may lead to delivery of entirely misconstrued information to BS and cause serious problems due to delivery of wrong information.

III. DESIGN OF SECURE PEGASIS (S-PEGASIS)

We propose a new protocol called secure PEGASIS which will be able to combat HELLO flood attack in PEGASIS. We chose CDMA based hierarchical PEGASIS as our base protocol as it is energy efficient as compared to other data centric or hierarchical protocols for WSNs. CDMA means code division multiple access method. It is used for parallel communication between multiple pairs of nodes using single communication medium (here the radio used for data transmission). Each pair uses a different 'code' to encode its data. The combined data from all senders is transmitted over the medium. Any node can receive the data sent by any other node by decoding the data using the sender's code. This way, no multiplexing or time sharing is needed and multiple pairs can communicate with each other simultaneously. The protocol is designed keeping in mind a WSN with the following assumptions:

1. The sensor nodes and BS are stationary.
2. The nodes are capable of communicating with BS but require extra energy for this. For communication within network, much low power radio is used so that energy reserve of nodes is not exhausted quickly.

1.1 Features:

We try to assure the following features in the proposed protocol:

1. Chaining concept of S-PEGASIS:

The nodes will be arranged in a chain with a leader transmitting to the BS and other nodes communicating only with one-hop neighbors. Nodes will be numbered 1 to N (where is the total number of nodes). This number has to be stored in memory. The nodes take turn in becoming leader. In the i^{th} round of communication, the node numbered i mod N becomes leader. The nodes locate themselves and form chains by broadcasting HELLO packets to all neighbors with a signal strength that is approximately equal to that required for communicating with neighbors in normal radio range (not with extra energy like that required for communicating with base station). The chain is formed using a greedy approach. Any node chooses the node from which it first receives HELLO packet as its neighbor along with network key (key negotiation is used for security as explained later). Chain formation has to be done initially when the network is set up or when some node dies. Data transmission process is same as in hierarchical PEGASIS. It is explained below. We assume a chain of 8 nodes c0-c7 arranged linearly in an arbitrary round of data transmission, say round 3. So, c3 is the current leader. The transmission hierarchy looks like the one depicted in Fig. 2.

Since, node c3 is in position 3 (counting from 0) on the chain, all nodes in an even position will send to their right neighbor. The transmissions c0->c1, c2->c3, c4->c5 and c6->c7 occur simultaneously using CDMA. Nodes that are receiving at each level rise to next level in the hierarchy. Now at the next level, node c3 is still in an odd position (1). Again all nodes in an even position will aggregate its data with its received data and send to their right. At this level, the transmissions c1->c3 and c5->c7 occur simultaneously. At the third level, node c3 is not in an odd position, so node c7 will aggregate its data and transmit to c3. Finally, node c3 will combine its current data with that received from c7 and transmit the message to the sink.

Now, in next round, node c4 becomes leader. First, the transmissions c1->c0, c3->c2, c5->c4 and c7->c6 occur simultaneously. The receivers c0, c2, c4 and c6 rise to the next level. Then, the transmissions c2->c0 and c6->c4 occur simultaneously. Then the receivers c0 and c4 rise to the next upper level. Transmission c0->c4 occurs and c4, the leader, sends the aggregated data to the BS. In normal PEGASIS, it would involve eight phases of data transfer in every round as no two pairs communicate simultaneously due to shared communication medium. But here, only 3 phases of data transfer are needed (corresponding to levels in the hierarchy) due to simultaneous data transmissions in the lower levels of hierarchy.

To analyze it mathematically, we consider a network with N nodes where$N = 2^m$ .In the lowest level of hierarchy, there are N nodes, out of which $N/_2$ are senders and $N/_2$ are receivers. The receivers rise to the next level in hierarchy. So, there are $N/_2$ nodes in the 2nd level. Similarly, there are $N/_4$ nodes in the next level and so on, till there is only one node, the current leader, in the topmost level.

We assume lowest level to be level 0, the next to be 1 and so on.

No. of nodes in level 0 = $N/_{2^0}$=N, no. of nodes in level 1 =$N/_{2^1}$=$N/_2$, no. of nodes in level 2 = $N/_{2^2}$=$N/_4$.

Let the topmost level be kth level. As no. of nodes in this level is 1, $N/_{2^k} = 1$.

Therefore, $k = \log_2 N$. As no. of levels = no. of phases in data transfer i.e, the maximum delay a node can have in relaying its data is $O(\log N)$.

In normal PEGASIS chain, as nodes are linearly arranged, data is passed from one node to other along the chain with only one node transmitting at an instant. So, delay for the farthest node is O (N). So, we can see that hierarchical chaining decreases delay. Since delay is the biggest factor affecting scalability of PEGASIS, it increases scalability.

2. Scalability:

The number of nodes in the network can be increased without causing much overhead or congestion as only one of them communicates to the base station and each node communicates with only one neighbor in each step (level of CDMA tree) of every round. The only overhead incurred will be in chain formation. But this is not done frequently, and since a greedy approach is used, it does not consume too much time. If a chain breaks or a node is corrupted by an attacker in such a way that it refuses to pass packets, its expected receiver neighbor will inform the BS about the non-responsiveness of the node after it fails to provide data in a given round. It does so by sending a special alarm packet. The BS will query and authenticate the node to check if the failure is permanent (node dead or corrupt). In this case, the chain will be altered to bypass this node. Otherwise, if it is due to low transmission power, the BS instructs the node to increase transmission power.

3. Static Nodes:

To avoid dynamic topology adjustment problem of LEACH, nodes are static which means that they transmit data to fixed neighbor only once a chain has been formed. Though the leader is changed in every round, it does not affect the transmission mechanism of other nodes. By making the nodes static, we restrict the overall energy consumption of the network.

4. Multi-Hop Communication:

The node receiving a packet from a neighbor aggregates it with its own packet to remove redundancy. There are no redundant packets transmitting in the network as aggregation is done as soon as a packet goes one hop. So, data volume transmitted is minimized and this increases energy efficiency.

5. Security:

We try to combat HELLO flood attacks by introducing key negotiation at the time of chain formation. Each node is supplied with a network key which is globally shared in the network and not known to any outsider. It is also provided with an algorithm to encrypt the key during negotiation.

The algorithm can be any encryption algorithm which is same for all nodes and known by the BS. Suppose, we are using a two byte network key, then the algorithm may be to multiply the lower byte by 7, discard carry and divide the higher byte by 3 to get the encrypted two byte key. Or the algorithm may be encrypting the key using another key. The algorithm is not protocol specific, which means, it may be different for different networks employing this protocol. The network designer chooses this algorithm according to convenience. It should not be known to any outsider (that is, devices excluding the BS, nodes and designer of the network).

The key and algorithm are stored in a special secure memory. In case key-based encryption is used, the algorithm may be stored in normal memory and only the key (used for encrypting the network key) needs to be stored in secure memory. Some secure memories are available for the purpose of storing secret keys and protected software. Their content cannot be accessed by any outsider. Only the device on which they are mounted can read them. One example is the Atmel’s crypto memory EEPROM. It can be seamlessly integrated into the node’s architecture and accessed just like its own memory. The implementation is done totally in hardware. Secure memories are costlier than normal memory but since we use it to store only the key and its algorithm, we need only a small amount of memory which would not affect the overall cost of nodes much.

While publishing HELLO packets, the nodes also have to supply the key. The key is encrypted so that any attacker cannot obtain it by eavesdropping. It is encrypted by applying the algorithm which is stored along with it in the secure memory.

If receiver finds a match with its own key after encrypting it, then only the sender is added in the chain. Any attacker causing HELLO flood is not expected to know this information and hence will not be added to the chain. During data transmission, the packets should again contain this key to ensure to any receiver node getting data from a neighbor that the sender is a legitimate node. This, along with the alarm raising mechanism in case of failure of a node to transmit data, makes an attack detection system. Sinkhole attacks, wormhole attacks and Sybil attacks are inherently absent in this protocol because attackers are not allowed to be a part of the chain at all. So, they cannot publish any route through them to other nodes. No sinkhole can be created. Attacker detection ensures that even if malicious nodes get some packets of the network and form a wormhole with their invisible band low latency link, the nodes will be detected as attackers as soon as they try to pass the packet to a legitimate neighbor as they cannot get the correct network key. In other words, base station never gets wrong or made up information.

1.2 Detailed Design:

1.2.1 Chain Formation:

The first step in this protocol is the formation of secure chain. The major steps in chain formation are:

- All nodes set their left and right neighbors as null.
- The BS sends a TOKEN packet to any randomly chosen live and non-malicious node.

- A node, on receiving a TOKEN packet, searches for a right neighbor. It broadcasts HELLO packets. To reduce number of packets, HELLO is sent only to live nodes which do not have a left neighbor.
- On receiving a HELLO packet, a node sends neighbor request NREQ packet to the sender in a bid to choose it as left neighbor. A node may send requests to more than one node on receiving HELLO from them.
- Along with the NREQ packet, the public key is also sent. The sender extracts the public key, encrypts the network key with this public key and sends a neighbor accept NACC packet with this encrypted key and its own public key. The network is unique for the network and not known to any outsider.
- When the request sender gets an NACC packet, it extracts the encrypted key, decrypts it with its private key and matches with the stored network key. If there is a match, this means the sender is legitimate. It is chosen as left neighbor and a confirmation packet NCONF is sent. This packet also contains the network key encrypted with the left neighbor's public key so that it can verify the legitimacy of this sender.
- When a node receives the NCONF packet, it verifies the network key. If it is correct, the sender is chosen as right neighbor and the token is passed to it using the TOKEN packet.
- This process repeats till a suitable right neighbor is found for all nodes except the rightmost node. That is, the chain is complete when the count of nodes with right neighbors is one less than the total number of valid nodes.
- After the chain is complete, all nodes are assigned positions (starting with 0 from the left). Positioning is important for the protocol program to work according to the hierarchical CDMA transmission model.

1.2.2 RSA Algorithm:

We have used public key cryptography for communication of network key. The network key, which is exchanged between neighbors during chain formation, is communicated in an encrypted form so that any outsider malicious node may not intercept it. Asymmetric key cryptography is used because it does not incur the overhead of key exchange, which is very costly for sensor network due to constraint on energy. We use RSA algorithm for key generation, encryption and decryption. The algorithm is described as follows:

- All nodes choose a public key and a private key. This is done using the following steps:
- Choose two unequal prime numbers p and q. Find their product n and a quantity ϕ given by (p-1)(q-1).
- The public key e is any number less than ϕ and co-prime to ϕ.
- The private key d is a modular inverse of e mod ϕ, that means, d*e (mod ϕ) =1.
- The public key is taken as e!n and private key as d!n.
- The nodes publish their public keys whereas the private keys are kept securely so that no outsider can know it. It is generally very difficult to guess the private key since finding the product of prime numbers is much easier than factorizing the product. Secondly, the values p and q are randomly chosen and e and d are found based on modular arithmetic.
- In order to send some data (in this case the network key), the sender encrypts it using the receivers' public key. The receiver decrypts is using its private key. It can be correctly decrypted if and only if it was encrypted using the corresponding public key.
- The algorithm used for encryption and decryption are same due to modular arithmetic. Either of the public and private keys may be used for encrypting and the other one can decrypt the encrypted data. This encrypting algorithm is:
- Represent the data in a numeric format.
- If d is the data, and e!n is the receivers' public key, the encrypted data will be d ^ e mod n.
- Similar method is used for decryption. If c is the encrypted data, the original data can be found by c ^ d mod n, where d is the private key of the receiver.

1.2.3 Data Transmission:

Data transmission in this protocol occurs using the hierarchical model described previously. This has been adopted to reduce delay and hence enhance scalability. The time complexity is O (log n) where n is the number of nodes. The major steps in this algorithm are:

- Data transmission occurs in rounds. BS issues the start round commands to the nodes. They transmit the data they have collected from surroundings, on receiving this command. Every round has a different leader, chosen by the formula i mod N, where i is the round number and N is the total number of nodes in the chain.
- Each round consists of phases. The first phase contains all the nodes. The nodes are divided in pairs. One in each pair transmits data and the other receives from it and aggregates it with its own data. The next phase consists of the receivers of first phase and a similar process is followed. These transmissions take place simultaneously using CDMA.

- The direction of data transmission i.e, which node is the sender among a pair and which node is the receiver, is decided by the position of leader among the participating nodes in the current phase. If leader is in odd position, transmission occurs from right to left and if it is in even position, the direction is left to right.
- The number of phases is given by log N where N is the total number of nodes in the chain. In the last phase, only the leader is left with the data aggregated from all the nodes. The leader sends it to the BS.

1.2.4 Energy Modeling:

We have first demonstrated the formation of chain and data transmission without considering the effect of energy on it. Then we have developed an energy model to represent loss of energy at every step. The network uses two types of radio - a low power radio for communication within the network and a high power radio for contention free communication with base station. This is necessary for maintenance of global variables. Since two kinds of channels cannot be simultaneously modeled using ns2, we have shown the low-power radio communication as it occurs (with all packets explicitly shown) whereas the global variable communication is modeled using global tcl variables without showing exact communication with the BS. However, we have incorporated the energy consumption due to updating of global variables. When a node updates a global variable, it is sent to the BS which in turn sends it to every other node.

During the protocol operation, energy is consumed for sending and receiving packets, as well as sensing. Energy required to send a packet is much more than that to receive. The sensing power is very low compared to these.

The nodes are initially set up with energy of 5000 Joules. Then, we have defined constants to represent consumption of energy per byte for various network packets. These, multiplied by the number of bytes in a packet, gives the total energy consumed for various network events like sending a packet to another node, sending a packet to BS, receiving packet from another node, receiving packet from BS, idle (sensing energy consumption every 10 seconds) etc. The number of bytes for various kinds of packets like HELLO, TOKEN, global variables, sensed data etc have also been defined as constants.

When a node sends or receives some packet or updates or receives a global variable, its energy is consumed. If the energy level of a node falls below a certain threshold, it is considered dead. Dead nodes cannot send, receive or sense data. The protocol has features to detect and bypass dead nodes in the network. At regular intervals of time, the nodes check if their neighbors are dead. In this case the next neighbor of the dead neighbor is taken as a neighbor and the global network parameters adjusted accordingly. During data transmission if a node cannot transmit data, it is treated as dead.

If the number of dead nodes exceed a certain percentage (here 10%) of the total number of nodes, the chain is dissolved and a new chain is formed. This is done to ensure that due to interspersed dead nodes, the neighbors do not become far apart.

1.2.5 Error Control Mechanisms:

The network assumed for this protocol is a wireless sensor network with ad-hoc deployment. So, there are frequent instances of packet loss. In order to ensure participation of all nodes in chain formation, we have incorporated some error control mechanisms.

Whenever node broadcasts HELLO packets, it schedules a retransmission after a fixed time interval in case it fails to get a right neighbor. This accounts for loss of HELLO or NCONF packets. NREQ packet loss has not been considered since requests are received from a large number of nodes, and re-sending incurs sending energy cost. Similarly, NACC packet loss is not considered as ultimately, the node being unable to find right neighbor, will re-broadcast HELLO.

If TOKEN packet is lost, it is detected through the maintenance of a global variable indicating the presence of token in the network. In this case, the detecting nodes resend TOKEN packet to their right neighbors. To avoid reception of duplicate token, a node is restricted to respond to a TOKEN packet only if does not have a right neighbor.

Loss of NCONF packet may lead to isolation of the sender, because it has already fixed its left neighbor, so is unable to respond to other nodes' HELLO packets. To avoid this, a node periodically checks if its left neighbor is alive and has right neighbor null till it is selected as the right neighbor of that node. Otherwise, it sets its left neighbor as null.

Data packets may also be lost or arrive out of sequence in which they were sensed. But this is not accounted since detection of lost data may lead to a large overhead.

IV. SIMULATION RESULTS

1.3 NS2 Implementation and Results:

We have implemented our protocol by extending the Agent/MessagePassing class of NS2. Our class name is SPEG. We have added many procedures and overridden the recv and has_data functions of the base class to incorporate the desired functionality.

When a neighbor sensed data, it is modeled through the has_data procedure. In this protocol, the agent attached to each node manages the data and control information stored, manipulated and transformed by the node. Each node has data buffer to hold sensed data. When a node senses new data, the previous contents are replaced by the latest data. Similarly, when it transmits data to a neighbor during transmission phase, its buffer is cleared. When it receives data from a neighbor, it aggregates the data with its own data.

We have used a simple aggregation and fusion model where data is appended to buffer if not present there and it is discarded if already present in the buffer. When a node does not have any data, its buffer is empty and on time o transmission, it will send empty DATA packet.

The recv function has been overridden to represent the course of action when a particular packet is received by the node. The node can respond only if it is not dead. It first decides the type of the packet whether HELLO, DATA etc. and then takes action accordingly.

Besides the basic networking framework, procedures have been added for implementing the RSA algorithm, error control, chain formation and position assignment and scheduling the data transmission timings. The basic working program for the protocol is implemented as an external procedure which schedules the data transmissions after the chain formation is complete.

Nam is a Tcl/Tk based animation tool for viewing network simulation traces and real world packet traces. It supports topology layout, packet level animation, and various data inspection tools. For example when a network with 6 nodes, the visualization tool nam will display a 6 node network.. The location of the nodes could have been chosen at random. If a random location of nodes is chosen and it is not satisfactory, one can chose the “re-layout” button and chose another location. One can also edit the location by clicking at the Edit/View button and then drag in each node to its required location by the help of the mouse. Other things that can be done in nam include coloring nodes, changing shape of the nodes, by default they are round, but they can be changed into square or hexagon etc, coloring links, adding and removing marks, adding labels: a label can appear on screen from a given time onwards, adding text: at the bottom frame of the NAM window, one can make text appear at a given time. This can be used to describe an event at that time. One can add in NAM, a monitoring of the queue file.

Network Animator shows animated results and analyses them in terms of the movement of various packets (TOKEN, HELLO, DATA etc.) in a network environment. Whereas ASCII trace can be used to analyze energy, throughput and various other parameters that determines the efficiency of the network.

1.4 Security and other features:

As the name suggests, the basic aim of this protocol is to add security features to PEGASIS. We have tried to combat various attacks on sensor networks. S-PEGASIS aims to remove all internal and external threats by incorporating only those nodes in the chain which are legitimate and part of the original network. This is done using key negotiation algorithm to combat HELLO flood attacks. As only legitimate nodes form the chain, other kinds of attack such as sinkhole and wormhole attacks are not possible because the path of packets in chain is fixed and cannot be influenced by an outsider malicious node. Sybil attack is also not possible because of unique numbering of nodes by the protocol. An outside node pretending to have a valid number cannot include itself in the chain due to lack of information about the network key. This protocol also ensures reliability in the network and hence especially suitable for critical data sensing networks. This is manifested in inclusion of each deployed node of the network in the transmission chain and immediate neighbor switching when a node is unable to transmit data.

Another important feature of this protocol which has been successfully implemented is the reduction in time complexity of data transmission. In a normal chain like in PEGASIS, time complexity of one round of data transfer (data collected by all nodes aggregated and sent to BS) is O (n), where n is the total number of nodes in the chain. In S-PEGASIS, it becomes O (log n) due to the hierarchical CDMA bases approach in which there are multiple transmissions in each phase of a round.

1.5 Comparative Study:

We have studied some existing hierarchical protocols – mainly LEACH and PEGASIS while developing the concept of S-PEGASIS. We have tried to incorporate the best features of these protocols into our design beside the essential feature of security. Here, in TABLE I, we present a comparative study of these protocols and their features along with our design of S-PEGASIS.

Table I:

Protocol / Parameter	LEACH	PEGASIS	S-PEGASIS
Scalability	It is scalable because of hierarchical data transmission.	It is not very scalable due to transmission time complexity of O (n) which causes large delay if chain is long.	It is scalable. The transmission time complexity is reduced to O (log n).
Energy Efficiency / Bandwidth Consumption	It is highly energy efficient compared to data centric protocols but its efficiency is less than single chain protocols due to a number of cluster heads communicating with the BS.	It is highly energy efficient as a single chain is formed with only one node called leader communicating with the BS and also because other nodes communicate with their neighbors only.	It is highly energy efficient as a single chain is formed with only one node called leader communicating with the BS. Energy consumed while forming the chain is a little higher than PEGASIS but overall efficiency is almost same.
Network Lifetime	Very Good	Very Good	Excellent
Multi-Path Routing	No	No	Yes
Classification	Clustering	Reactive / Clustering	Clustering / Hierarchical
Data Aggregation	No	Yes	Yes
Security	It does not provide security.	It does not provide security.	It has in-built security mechanism. Only legitimate nodes can become a part of the chain.
Reliability	It does not guarantee inclusion of every node in data transmission. If a node fails to respond to cluster head's advertisement, it is left out.	It does not guarantee inclusion of every node. A node relatively farther from other nodes but within radio range may be excluded from chain.	It guarantees inclusion of each live and non-malicious node within radio communication range of others, in the chain.

V. CONCLUSION

Sensor Networks hold a lot of promise in applications where gathering sensing information in remote locations is required. It is an evolving field, which offers scope for a lot of research. Their energy-constrained nature necessitates us to look at more energy efficient design and operation. Protocols, which name the data and query the nodes based on some attributes of the data are categorized as data-centric. Many of the researchers follow this paradigm in order to avoid the overhead of forming clusters, the use of specialized nodes etc.

On the other hand, cluster-based routing protocols group sensor nodes to efficiently relay the sensed data to the sink. The most interesting research issue regarding such protocols is how to form the clusters so that the energy consumption and contemporary communication metrics such as latency are optimized. The factors affecting cluster formation and cluster-head communication are open issues for future research. Moreover, the process of data aggregation and fusion among clusters is also an interesting problem to explore.

Our study of routing protocols was based on SPIN, LEACH and Hierarchical-PEGASIS, as these protocols best handle the issues of energy-awareness, security concerns and resource constraints, eliminating problems of implosion and resource-blindness found in the traditional routing protocols. Then we have developed a new protocol called S-PEGASIS which adds security to the existing PEGASIS hierarchical model.

This paper deals with creating a new protocol called Secure PEGASIS (or S-PEGASIS) for wireless sensor network that will be able to effectively use the limited energy and bandwidth, secure the delivery of data packets among various nodes, counter HELLO flood attacks, and at the same time enhance the scalability and reduce delay in transmission with the help of hierarchical PEGASIS.

The simulation, however, has certain limitations. For example, the transfer of global variables has been modeled through global variables of tcl. It can be improved to show exact packet transfer using a separate high power channel for communication of nodes with BS. Secondly, the protocol has a chain formation time complexity of O (n^2). As seen from the observation, a 512 node network will take approximately 10 hours to establish the chain. This is a practical limit on the number of nodes. So, there is scope of work in developing a new neighbor searching procedure with less complexity but at the same time ensuring the key negotiation and inclusion of each node on the chain.

We have not provided encryption for data due to large overhead of transmission of encrypted data (since data is huge in amount and asymmetric key cryptography produces cipher texts larger than the original text). Some more efficient encryption technique may be developed to tackle this situation so that the data being passed could not be traced by any outsider. It can be further extended by providing link layer and physical layer security mechanisms so that trusted nodes on chain can never be hacked.

In the generic context, the sensor networks can be made more robust by using solar cells on nodes, incorporating VLSI based memories with low cost and large storage etc. This protocol can be immensely beneficial in sensor network applications in critical area viz. battlefields, disaster management, nuclear radiation etc. where security is and reliability is necessary.

VI. REFERENCES

Journal Papers:

[1] Akyildiz, Ian F., et al. "Wireless sensor networks: a survey." *Computer networks* 38.4 (2002): 393-422.

[2] Akkaya, Kemal, and Mohamed Younis. "A survey on routing protocols for wireless sensor networks." *Ad hoc networks* 3.3 (2005): 325-349.

[3] J. N. Al-Karaki, A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” *IEEE Wireless Communication, vol. 11, issue 6*, pp. 6-28, 2004.

[4] Y. Wang, G. Attebury, B. Ramamurthy, “A survey of security issues in wireless sensor networks,” *IEEE Communications Surveys and Tutorials, vol. 8, issue 2*, pp. 2-23, 2006.

[5] Bhattacharyya, Debnath, Tai-hoon Kim, and Subhajit Pal. "A comparative study of wireless sensor networks and their routing protocols." *Sensors* 10.12 (2010): 10506-10523.

Proceedings Papers:

[6] Chris Karlof, David Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures,” *Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 113-127, May 2003.

[7] Lindsey, Stephanie, and Cauligi S. Raghavendra. "PEGASIS: Power-efficient gathering in sensor information systems." *Aerospace conference proceedings, 2002. IEEE*. Vol. 3. IEEE, 2002.

[8] Acs, Gergely, Levente Buttyán, and István Vajda. "Modelling adversaries and security objectives for routing protocols in wireless sensor networks."*Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*. ACM, 2006.

[9] Buttyán, Levente, and István Vajda. "Towards provable security for ad hoc routing protocols." *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*. ACM, 2004.