MS EE Scholarly Paper
Spring, 2009

Secure Routing in Wireless Sensor Networks

Srividya Shanmugham
Scholarly Paper Advisor: Dr. Jens-Peter Kaps

George Mason University
Fairfax, VA

Abstract— **Wireless sensor networks (WSN) is an emerging area that has a wide spectrum of critical applications like battlefield surveillance, emergency disaster relief systems, etc. Sensor devices used in such networks are designed to operate with limited resources. Therefore, they are simple to build, economically viable and can be deployed to closely interact with their environment. In order to reduce the amount of data to be transmitted they perform in-network processing by aggregating useful information. These characteristics of sensor networks pose unique challenges for routing data securely over wireless communication channels. Traditional security techniques cannot be adopted easily due to resource constraints. Security in WSNs can be properly addressed only by integrating secure data transmission into the routing process itself. In this paper we outline different routing attacks in WSN and discuss how various sensor network routing protocols breakdown in the face of those attacks. We then list a set of attributes that would make a routing protocol more secure. Finally we study a new protocol called Secure Sensor Network Routing Protocol [17] that was designed to be resilient to routing attacks and analyze the strength of its security.**

I. Introduction

Advances in digital electronics and wireless communication technologies have led to the development of tiny, low-powered, low-priced devices called sensor nodes. They consist of sensing, data processing and communicating components as shown in Figure 1.

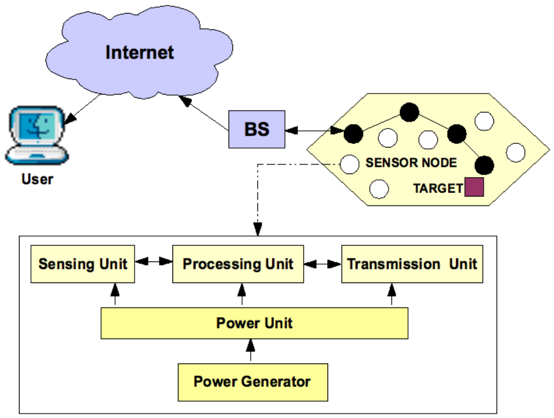

Fig. 1. Wireless Sensor Network Architecture

Interconnections of sensor nodes over an area form an ad-hoc, infrastructure-free, multi-hop network called Wireless Sensor Network (WSN) [1]. Sensor nodes interact closely with the physical environment in which they are deployed. They collect information about their surroundings and route the data towards Base Stations (BSs) through neighboring nodes. A Base Station is a fixed or mobile node connecting the sensor network to an external network. They help in either propagating control information into the WSN and extracting information from it. Typically a BS plays the role of a gateway to another network.

Sensor networks have critical applications like battlefield survelliance [2], emergency response systems [3], seismoacoustic systems for monitoring volcanic activity [4], RF-Based location tracking systems [5], Health and disaster aid system [6] etc. In such applications existing wireless communication routing protocols like TinyOS Beaconing, Directed Diffusion, etc. assume a trusted environment [7]. Therefore, communication amongst sensor nodes is susceptible to routing attacks like spoofing, selective forwarding, malicious packet injection, etc. Hence, the need for secure sensor networks.

Routing protocols should be secure enough to enable communication despite malicious activities. But constrained capabilities of sensor nodes in terms of bandwidth, energy supply and computational resources complicate routing security. Traditional expensive security techniques are not suitable for sensor networks. Security in WSNs can be achieved only by designing and building secure routing protocols specially suitable for WSNs.

In Section II, we discuss different routing attacks that can be mounted on a sensor network. In Section III we analyze the level of security that existing routing protocols [8] of WSNs can provide. After knowing how and why routing attacks in WSNs work, in Section IV we present a list of attributes that should be taken into account while building a security system for WSNs. In Section V, we study a new protocol called Secure Sensor Network Routing Protocol that was designed to provide security in resource constrained devices. Finally we analyze the security of this protocol and provide some concluding remarks.

II. Routing Attacks in WSNs

Wireless communication in WSNs is insecure because an adversary can easily eavesdrop into the radio transmissions or replay overheard transmissions. A mote-class attacker can pretend to be an ordinary node and can jam the radio link in its vicinity. A laptop-class attacker with more battery power, sensitive antenna, high bandwidth can bring down the entire network. From security point of view WSNs are weak at the routing layer. In this section, we shall discuss the mechanisms deployed by attackers for mounting a variety of attacks aimed at eliminating important routed data, causing flow suppression, reducing network performance and modifying the routing tables in the nodes of a sensor network.

A. Selective Forwarding Attack

In general every node in a WSN trusts its neighboring nodes to forward its packets to the next hop. By becoming a part of this trust model, a malicious node can simply refuse to forward data packets to the next hop. By doing so, after a period of time the nodes in the vicinity of malicious node might consider it to be a normal node with operational defect. This would make them build new routes by avoiding the malicious node. In order to prevent being neglected by neighboring nodes, the malicious node selectively drops the packets from few of its neighboring nodes. This approach limits the suspicion of wrongdoing by the malicious node and ultimately succeeds in disrupting the routing set up in the network.

B. Spoofed or Replay Attack

This kind of attack targets the application data that is being routed between nodes. For example, in an inventory application, a competitor should not have access to the inventory data communicated across the network. By spoofing or altering or replaying such routed information, false messages can be generated, routing loops can be created, latency of the network can be increased, etc. The motivation for mounting a replay attack is to encroach on the authenticity of the communication in WSNs.

C. Sinkhole Attack

In the network model of WSNs, a Base Station(BS) is the final destination for all the nodes. Routing algorithms usually help nodes in finding the best possible route to BS. By providing a high quality route to base station compromised nodes can easily become a sinkhole. In this type of attack, adversary's main aim is to attract almost all the traffic from a particular area towards the compromised node as shown in Figure 2. For instance, using a powerful transmitter a laptop-class adversary can reach the BS in a single hop thus creating a sinkhole in the network. The nodes that forward their packets through the sinkhole, propagate information about the quality of the sinkhole to their neighboring nodes. More nodes are thereby attracted towards the sinkhole and the size of group increases gradually until the entire network is covered. Routing protocols that verify the quality of the route based on end-to-end acknowledgment also fall prey to this kind of attack.

One motivation for launching a Sinkhole attack is that it enables selective forwarding attacks to be carried out easily. After establishing itself as a sinkhole, an adversary gets an opportunity to selectively suppress or modify the packets received from other nodes in that area.

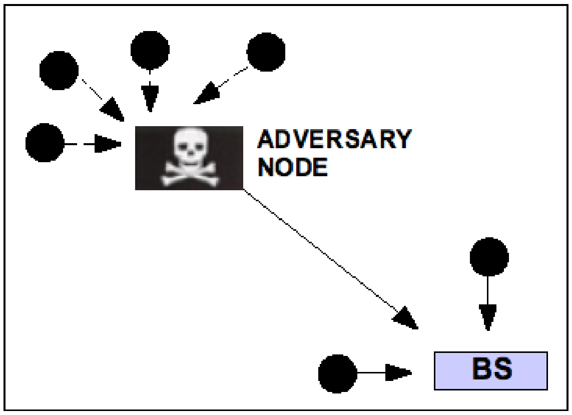

Fig. 2. Sinkhole Attack

D. Sybil Attack

Every node in WSNs has a routing table of limited size filled with routing identites of its neighbors [9]. A node makes routing entries based on the quality of the route that can be supported by its neighboring nodes to reach the BS. The size of the table is checked before making new entries. If the table hasn't reached the maximum limit, entries are made in the order of arrival of the identity messages from neighboring nodes. If the table is full, entries in the table are compared with the newly arrived entry for the quality of the route. The node with least quality in the table is replaced by the new node with a better quality.

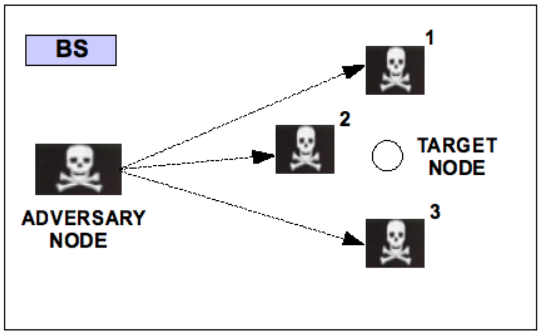

Fig. 3. Sybil Attack

In this kind of attack, an adversary node takes up various identities and locates itself at different places inside the network as shown in Figure 3. With different identities it fills out the routing table of a sensor node. By doing so, the sensor node is excluded from the rest of the network. The downside of achieving this goal is that the attacker requires a huge amount of data. This makes the attack resource intensive. Another reason for presenting multiple identities by a single adversary is to reduce the effectiveness of fault tolerant schemes like multipath routing, distributed storage and topology maintenance.

E. HELLO Attack

Nodes in WSNs learn about their neighboring nodes through *HELLO* packets. Every node advertises its presence to neighboring nodes by broadcasting *HELLO* packets. A malicious node follows the same technique. It uses transmission power high enough to reach the nodes that are very far away from its physical location which convinces the receivers of its advertised packets that it is a legitimate neighboring node as shown in Figure 4. Generally routing protocols of WSN depend on localized exchange of routing information to maintain routing topology and flow control.

The main motivation for carrying out *HELLO* Attack is to perturb topology maintenance thereby leaving the network in a state of confusion.

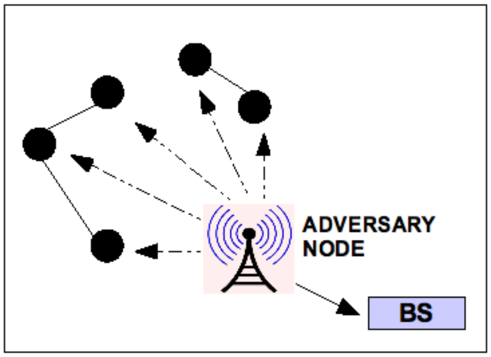

Fig. 4. HELLO Attack

III. Analysis of routing protocols of WSNs

Analyzing routing protocols from the security perspective can help in enumerating the defects in them. This in turn can help in determining security attributes that should be taken into consideration while building secure routing protocols. In this section we discuss the working principle of routing protocols and their vulnerability to different attacks [10].

A. TinyOS Beaconing

This protocol builds a spanning tree with a base station as the parent for all the nodes in the network. The base station broadcasts route updates periodically to its neighboring nodes which in turn broadcasts it to their neighboring nodes. This process continues recursively with every node marking its parent node as the first node from which it receives the route update for the current time epoch.

The simplicity of this protocol makes it susceptible to all the attacks discussed in the previous section. Broadcasting unauthorized routing update is possible that makes spoofing of data easy. Any node can easily claim to be the base station and can become the parent of all nodes in the network. A laptop class adversary can carry out HELLO flood attacks by transmitting a high power message to all the nodes and by making every node to mark the adversary as the parent node.

B. Directed Diffusion

Directed Diffusion is a data centric protocol. The base station queries for data by broadcasting interests. An interest describes a task required to be done by the network. The intermediate nodes keep propagating the interests until the nodes that can satisfy the interests are reached. Every node that receive the interests sets up a gradient toward the node from which it received the interest. A gradient mentions an attribute value and direction. As shown in Figure 5 when node B receives an interest from node A, it includes A(Δ) in its gradient. When node C receives an interest from node A through node B, it includes B(2Δ) in its gradient. When node C receives an interest from node A, it includes A(Δ) in its gradient. The data generated by the sensor nodes would be named as attribute-value pairs. When the data matches the interest(event), path of information, flows to the base station at low data rate. Then the base station recursively reinforces one or more neighbors to reply at a higher data rate.

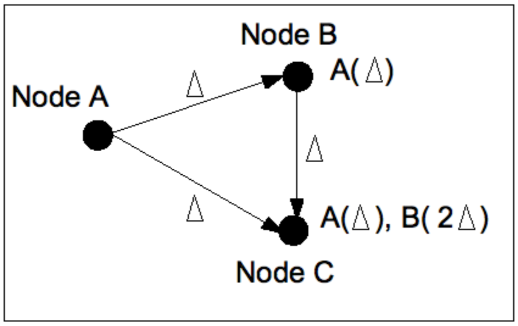

Fig. 5. Gradient set up in Directed Diffusion Routing Protocol

In this protocol it becomes an easy task for the attacker to eavesdrop the interest. When interests are broadcasted by the base station, the adversary can also receive it and can forward it to other nodes. When the response for that interest is sent, apart from the base station, the adversary would also be receiving them. By spoofing positive and negative reinforcements, the adversary can easily influence the path through which responses are being sent back to the base station. This helps the adversary to modify and selectively forward the data events.

C. Geographic and Energy Aware routing [GEAR]

This protocol uses energy aware and geographically informed neighbor selection heuristics to route a packet toward the destination region. Every node maintains two different costs for reaching a destination through neighboring nodes: (i) An estimated cost that is a combination of residual energy in the battery of the node and its distance to the destination; and (ii) A learning cost that accounts for routing around holes in the network. A hole is formed when there is no other node closer to the target other than itself. On receiving a packet, a node checks if any of its neighbors is located closer to the target region. If there are more than one, the one that is closest to the target is chosen. If there is only one of its neighboring nodes, it choses that node. If there isn't any, then there is a hole and it picks one of its neighbors to forward the packet based on the learning cost function.

Attacks can be launched by an adversary node by just advertising to have maximum energy. An adversary can carry out Sybil attack by covering up the target node with multiple bogus nodes. After taking part in the routing process, it can carry out selective forwarding attack.

D. Rumor routing

Rather than flooding the entire network to retrieve information about events(data matching the query), as in directed diffusion protocol, this protocol uses long lived packets called agents. When a source node observes an event it generates an agent. Agents travel the whole network and propagate information about the local events to distant nodes. They carry a list of events, next hop path to those events, hop count of those paths, a list of previously visited nodes and a Time To Live (TTL) field. On arriving at a new node the agent informs that node about the events it knows of and adds to its event list any event the node might know of. It decrements its TTL field. If TTL is greater than zero the node probabilistically chooses the agent's next hop from its neighbors present in the routing table minus the previously visited nodes listed in the agent. Similarly base station creates an agent to propagate the query into the network. When an event agent arrives at a node previously traversed by a query agent querying for the event, a route from base station to source is set up.

This protocol is dependent on nodes forwarding the agents properly. By just removing the event information carried by the agent or by refusing to forward the agent an adversary can carry out denial of service attack. Laptop class attackers can carry out Sybil attacks and selective forwarding attacks.

E. Minimum Cost Forwarding Algorithm [MCF]

According to this protocol, the sensors don't have to maintain routing tables or have unique IDs. Instead, every node maintains the least cost estimate from itself to the base station. Least cost estimate is calculated according to distributed shortest path algorithm. Each message to be sent to the base station is broadcasted by the nodes. The neighboring nodes that receive this message determine if they are on the least cost path between the sender and the base station. If so, then they would rebroadcast the message. An adversary node can act as a base station by advertising to have zero cost. It becomes the sole destination of all the nodes in the network. This is possible especially when a laptop class attacker sends HELLO messages. In a HELLO message the adversary node can advertise to have zero cost with transmission energy powerful enough to be reached by all the nodes in the network.

F. LEACH protocol

LEACH is a hierarchical clustering algorithm for sensor networks. This protocol randomly selects a sensor node based on its received signal strength to be a Cluster Head (CH) as shown in Figure 6 and rotates this role to other nodes in the cluster. This distributes the energy load amongst sensor nodes uniformly. A Cluster Head compresses the data arriving from the nodes in its cluster. This reduces the amount of information transmitted to the base station. The operation of this protocol involves 2 phases: *Set up phase* and *Steady State phase*. During *Set up phase*, the clusters are organised and CHs are selected. In *Steady State phase*, actual data transfer to base station takes place. Sensor nodes start sensing and transmitting their data to CHs. After receiving all the data from its cluster, the CH node aggregates it and sends it to the base station. After some time, the network again goes to *Set up phase* and a new CH is selected.

In order to compromise the LEACH protocol an attacker attempts to assume the role of CH. Since a CH is chosen probabilistically it is difficult for an attacker with

TABLE I

SUMMARY OF ATTACKS ON ROUTING PROTOCOLS IN WIRELESS SENSOR NETWORKS

Routing Protocols	Selective Forwarding	Spoofed Attacks	Sybil Attacks	SinkHole Attacks	HELLO Attacks
TinyOS Beaconing	✓	✓	✓	✓	✓
Directed Diffusion	✓	✓	✓	✓	✓
GEAR	✓	✓	✓	-	-
Rumor routing	✓	✓	✓	✓	-
MCF	✓	✓	-	✓	✓
LEACH protocol	✓	-	✓	-	✓
GAF	✓	✓	✓	-	✓

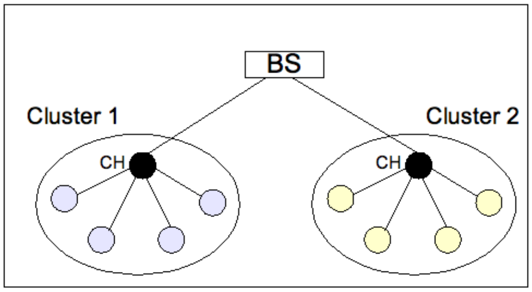

Fig. 6. LEACH Routing Protocol

a single identity to achieve this goal. However as in the case of Sybil attack where the attacker assumes multiple identities simultaneously, it is more likely for the attacker to be selected as a CH. Since the probability of a node being chosen as a CH is proportional to its received signal strength, a laptop class attacker can easily assume the role of a CH by using *HELLO* flood attack. After becoming a cluster head it can carry out the selective forwarding attack.

G. Geographic adaptive fidelity [*GAF*]

This is an energy aware location based routing algorithm. The network region is divided into fixed zones to form a virtual grid. Every node can be in any of the three modes: *Discovery*: to determine the neighboring nodes in the grid; *Active*: to reflect participation in the routing process; *Sleep*: to turn off the radio. Inside each zone the nodes colloborate to have only one active node and the other nodes of the zone remain either in discovery state or sleep state. For example, a node in active state in a zone stays awake. It senses and reports about the environment of the zone on behalf of other nodes that are in sleeping state. This helps in conserving energy by turning down the radio of unnecessary nodes without affecting the routing process of the network. Nodes are ranked according to their expected lifetime and current state.

When a node is in discovery or active state and when it hears a discovery message from a higher ranked node inside a grid, it goes to sleep state and after some time it returns to discovery state. So an adversary can impersonate a higher ranked node in a grid which in turn results the nodes of that grid remain in sleeping state. Also, when a laptop class attacker carries out HELLO and Sybil attacks by sending strong transmissions of faked discovery messages, the entire network can be put in sleep state for a long time.

IV. ATTRIBUTES OF SECURE ROUTING PROTOCOLS

Routing protocols of sensor networks discussed in previous section focus mainly on optimized utilisation of the resources. They were not designed with security in mind [11]. They assume to operate in trusted environments. This makes the routing protocols vulnerable to a variety of attacks. In order to make them secure, design process of such protocols should involve three main requirements: *Prevention, Detection/Recovery and Resilience*. *Prevention* requirement can be achieved through cryptographic mechanisms like authenticity verification, data confidentiality and data integrity. This helps in protecting the network resources from unauthorized nodes. *Detection* requirement enforces real time monitoring of the protocol participants. On detection of malicious behavior, *Recovery* actions should be triggered that help in eliminating the malicious participants. *Resilience* requirement ensures certain level of availability even in the presence of compromised nodes inside the network. In this section we discuss these three design principles in detail [12].

A. Prevention against attacks

1) Authenticity verification: Authenticity verification involves checking the accuracy of the origin of data –

source authentication. This helps in preventing attacks from outsiders and insiders.

By periodically verifying the identities of the nodes, outsiders with false identities can be prevented from entering the network. By adopting encryption techniques using a globally shared key, the network can be made secure against Sybil class of attacks. The global symmetric key is also more affordable for the energy starved sensor nodes unlike the costlier digital signature using public key cryptography.

However, when a compromised node within the network runs malicious code or when an adversary steals the key material from an authorized node, insider attacks can be carried out. By sharing unique symmetric keys between every node and base station the network can be made resistant to such attacks. When any two nodes want to communicate, they establish a shared key between them after verifying their identities with the help of base station.

2) Data confidentiality: Sensors communicate sensitive information that shouldn't be eavesdropped. This can be avoided by encrypting the data before transmission using symmetric cryptographic techniques.

3) Base station decentralization: When a base station acts as a sink for almost all the routing messages it makes the entire network vulnerable to attacks targeting base stations. As in geographic routing protocols it is always good to build the routing topology based on localized interactions without the involvement of the base stations.

B. Detection of attacks

1) Data Freshness: Data freshness ensures that the data delivered to the receiver is recent and it is not an old message being replayed by an adversary. By including counter values or by including a nonce in the packets the freshness of the data can be verified.

2) Topology Structure restriction: The structure of the topology should not expand randomly. If the network size is restricted to manageable size, then each node after being deployed would let the base station know about its neighbors and its geographic location. The base station can map the topology of the entire network. If any node goes down, it can be easily tracked. If there is a major change in the topology, the base station can become suspicious about some node being compromised inside the network or some attack with a laptop opening a sinkhole.

C. Resilent to attacks

1) Multipath transmission: Messages routed over n disjoint paths, with n compromised nodes can provide sufficient protection against attacks like selective forwarding. But having n disjoint paths is practically difficult in real networks. So braided paths that have common nodes but not common paths can provide sufficient protection against selective forwarding attacks. While forwarding packets, if every node can select their next hop probablistically from the routing table entries, attacker's task of predicting the routing path to the base station would become difficult.

V. Secure Sensor Network Routing Protocol

Enforcing security in existing routing protocols through public key cryptographic mechanisms would either make them more complex or would drain the resources of tiny sensor devices. Hence many secure routing protocols adopt symmetric key cryptographic mechanisms to provide security. But they do not provide complete security because they consider only few of the design principles. For example, SPINS [13] and TinySec [14] focus only on *Prevention* principle. They provide inadequate security in the presence of compromised nodes. As a preventive measure Secure Implicit Geographic Forwarding (SIGF) protocol [15] chooses next hop dynamically and non-deterministically rather than maintaining routing tables. Intrusion-Tolerant Routing protocol for Wireless Sensor Networks (INSENS) protocol [16] adopts multipath technique in order to make the network resilent to attacks. None of the proposed symmetric key based routing protocols incorporate all the three design principles: *Prevention, Detection/ Recovery and Resilience*. Hence the need to design and build a new protocol from scratch that would consider all the requirements as discussed in the previous section.

With security and efficiency as the central design parameters a new asymmetric key based routing protocol named 'Secure Sensor Network Routing Protocol' has been designed by *Parno et al.* [17]. The overhead and complexity of cryptographic mechanisms has been observed to be within acceptable limits.

A variety of techniques like secure neighbor discovery protocol, recursive grouping algorithm, grouping verification tree algorithm, honeybee technique are being used to design this protocol. To handle interference from even active attackers, routing tables and network addresses for each node are dynamically established. In this section we shall discuss different techniques of this protocol in detail.

A. Secure neighbor discovery protocol

All the nodes in WSN are assigned a unique ID by a network authority(NA) and they are installed with NA's public key and certificate(ID of a node signed using private key of NA). At the beginning of the routing process, every node is designed to be the only member of its own group. Nodes start to learn about their neighbours using secure neighbor discovery protocol. They broadcast their IDs and certificates. Nodes that receive the certificates verify them using the NA's public key. On successful verification the nodes that sent the certificates are added to the routing tables. The neighbor discovery protocol is designed to be time bound so that all the nodes should advertise themselves only during certain periods of time. After the conclusion of the neighbor discovery protocol there will not be any acceptance of new nodes in the network. Since this protocol doesn't require broadcasting of HELLO packets to learn about the neighboring nodes HELLO attacks cannot be launched. In addition this protocol prevents an adversary from introducing sybil nodes inside the network. Adversary node is prohibited from injecting a manufactured ID, since it will be unable to produce a proper certificate to match it. A compromised node is also prevented from altering its ID without invalidating its signature.

B. *Recursive grouping algorithm*

The main idea of this algorithm is to populate the routing tables of the nodes by merging groups of nodes into larger ones. A group *G*, sends a merge proposal to another group *G'*. If *G'* agrees to merge with *G*, both will merge to form a larger group as shown in Figure 7. If not, *G* will look for another group. After the merging process, each node in *G* adds an additional bit to its network portion of its ID or address in order to differentiate itself from other nodes of group *G'* as shown in Figure 8. Nodes of *G'* also does the same. Routing tables are updated with the entries of neighboring nodes that are in the path to the other group. Every group is assigned an unique ID which can be authenticated by GVT algorithm(see Section V-D) along with the size of the group. This process of merging continues until the entire network forms a single group.

By the end of recursive grouping algorithm, each node is assigned a unique network address, a routing table of next hop neighbors and a merge table that helps in authentication of each merging group. Deterministic functioning of this algorithm based on the size and IDs of the group prevents an adversary from injecting, altering or dropping packets.

C. *Resilient forwarding*

The routing table of a node guides in forwarding the packet to a group containing the destination address. When node *A*, wishes to send a packet to another node *B*, node *A* compares the most significant bit of the address of node *B* to that of its own address. If the bits match, node *A* keeps tracking down to the group to which *B* belongs to. If not, node *A* looks at its routing table to find out the nearest neighbor of node *B* which could forward the packet to node *B*. Since routes are not chosen based on advertised distances, Sinkhole attacks are prevented from being launched.

In addition to this basic forwarding mechanism high availability of message delivery can be achieved by multipath forwarding. When two groups *G* and *G'* merge, each node enters the ID of the neighboring node through which it heard about the other merging group. The edge nodes have different neighboring nodes to reach out the same merging group. Instead of storing just one neighboring node to reach that group, three different neighboring nodes can be entered in each entry of the routing table. This provides a node with an option of choosing the next neighboring node when there is failure in receiving acknowledgement from the recipient or when there is increased latency. This makes the protocol resilent to natural problems in the network. If a malicious node acts as an edge node, it may fail to announce the neighboring group about its own group by selectively dropping packets from its group. This malicious activity is overcome by having redundant edge nodes that take over the function of notifying the neighboring groups. This leads to the removal of malicious node from the routing tables of internal nodes of the groups.

D. *Grouping Verification Tree (GVT)*

GVTs are hash trees formed for every group with sensor nodes as leaves of the tree. Hash of IDs of nodes

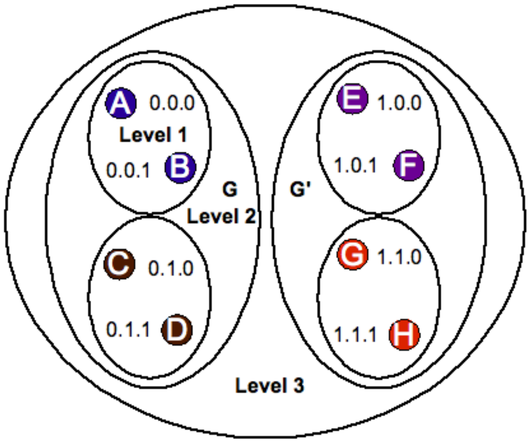

Fig. 7. Recursive Grouping Mechanism

Algorithm 1 GVT Verification Algorithm performed by Group *G* before merging with Group *G'*

```
$G$ chooses one of its nodes as challenger to verify the authenticity of Group $G'$. Prior to merging Group $G'$ announces its Group ID and size to Group $G$
Challenger broadcasts its challenge to its group $G$. Authenticity of the challenger is verified by Group $G$.
Edge nodes of Group $G$ forwards the challenge to Group $G'$
Based on the challenge value Group $G'$ chooses the responder from its group
$G$ chooses one of its nodes as challenger to verify the authenticity of Group $G'$
Responder sends its merge table and certificate to Group $G$
Nodes in Group $G$ carry out VerifyTree operation to verify the authenticity of Group $G'$.
```

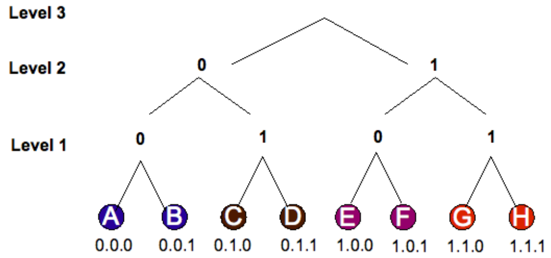

Fig. 8. Network Address of nodes

in a group refer to group ID. In Figure 7, nodes A and B merge to form a group ID V_{AB} at Level 1. Likewise nodes C and D merge to form group ID V_{CD} at Level 1. At Level 2, both these groups merge to form a group ID, V_G.

$$V_G = H(V_{AB}, V_{CD}) \tag{1}$$

$$V_G = H(H(ID_A, ID_B), H(ID_C, ID_D)) \tag{2}$$

Group *G'* also forms its group ID $V_{G'}$ in the same way. Each node in the tree has a merge table. Merge table records the ID and size of each group the node has merged with. For example, prior to merging of group *G* and *G'*, node *C* of Group *G* in Figure 7 would have two entries: *M[0] = [*ID$_D$*, 1]*; *M[1] = [*V$_{AB}$*, 2]*.

GVT helps in authenticating the merging groups by verifing their group size and group ID. When *G* and *G'* are about to merge, group *G'* announces its group ID and its size. *G* will select a challenger from its group based on its group ID. The node whose network address is prefix of H(V_G) is chosen as the challenger. The challenger has to prove its authenticity by sending its certificate to other nodes of the group. The challenger generates an authenticated challenge and is forwarded to *G'*. Based on this challenge, a responder is selected by *G'*. The responder provides its own ID, certificate and its merge table. The cerificate authenticates the responder's ID and nodes in *G* verifies the authentication of the responder. Merge table contains all the intermediate values in GVT of *G'*, group ID and group size. The challenger asks the responder to prove the existence of intermediate nodes of GVT. The responder gives the values of intermediate nodes such that the challenger can verify the group ID using them. Using *VerifyTree Authentication* operation [17] of hash tree mechanism, the challenger verifies the authenticity of *G'*. If there is any deviation in GVT, merging is aborted and next new group would be proposed for merging.

GVT helps in detecting malicious tampering during various levels of merging process. Any attempt to alter information about group IDs and size will be detected by *VerifyTree* operation of GVT. Deterministic choice of challenger and responder prevents an adversary from interrupting these choices. The challenge value can be calculated only by the challenger node and can be verified by other nodes of its group. The response value is verified by both groups. Hence there is no possibility of a malicious node to be chosen as either challenger or responder.

E. *HoneyBee Technique*

Presence of a malicious node can bring down the entire network easily. This can be avoided by isolating the malicious node. When a legitimate node detects a malicious node by any of the techniques described above, immediately it floods the network with the malicious node's id, its own id and its certificate. On receiving this notification the other nodes revoke both the nodes(legitimate as well as malicious nodes). Even if a malicious node claiming to be a legitimate node points out a legitimate node as malicious one both the nodes would be revoked by the entire network.

VI. Conclusion

Wireless Sensor Networks would be widely deployed in future mission-critical applications. But security problems at routing layer have to be resolved before their deployment in real world situations. A secure routing protocol should possess preventive measures against known attacks. On detection of any suspicious activity of a malicious node recovery mechanisms should be triggered. Stability of the network should not be drastically disturbed even in the presence of the malicious node. Current routing protocols are built for trusted environments. Enforcing security in existing routing protocols would make them complex. Secure Sensor Network Routing protocol provides good security against all known attacks. On implementing this protocol in TinyOS environment on testbed of Telos motes, it has been observed that the performance overhead is within acceptable limits compared to the level of security achieved.

References

[1] A. Ali and N. Fisal, "Security enhancement for real-time routing protocol in wireless sensor networks," *Wireless and Optical Communications Networks, 2008. WOCN '08. 5th IFIP International Conference on*, pp. 1–5, May 2008.

[2] E. Shi and A. Perrig, "Designing secure sensor networks," *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol. 11, no. 6, pp. 38–43, 2004. [Online]. Available: http://dx.doi.org/10.1109/MWC.2004.1368895

[3] T. Gao, C. Pesto, L. Selavo, Y. Chen, J. G. Ko, J. H. Lim, A. Terzis, A. Watt, J. Jeng, B. rong Chen, K. Lorincz, and M. Welsh, "Wireless medical sensor networks in emergency response: Implementation and pilot results," *Technologies for Homeland Security, 2008 IEEE Conference on*, pp. 187–192, May 2008.

[4] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *Internet Computing, IEEE*, vol. 10, no. 2, pp. 18–25, March-April 2006.

[5] K. Lorincz and M. Welsh, 2005.

[6] T. Gao, T. Massey, L. Selavo, D. Crawford, B. rong Chen, K. Lorincz, V. Shnayder, L. Hauenstein, F. Dabiri, J. Jeng, A. Chanmugam, D. White, M. Sarrafzadeh, and M. Welsh, "The advanced health and disaster aid network: A light-weight wireless medical system for triage," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 1, no. 3, pp. 203–216, Sept. 2007.

[7] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: a survey," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 6–28, Dec. 2004.

[8] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pp. 113–127, May 2003.

[9] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis and defenses," *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, pp. 259–268, April 2004.

[10] D. Wu and G. Hu, "Research and improve on secure routing protocols in wireless sensor networks," *Circuits and Systems for Communications, 2008. ICCSC 2008. 4th IEEE International Conference on*, pp. 853–856, May 2008.

[11] G. Niezen, G. Hancke, I. Rudas, and L. Horvath, "Comparing wireless sensor network routing protocols," *AFRICON 2007*, pp. 1–7, Sept. 2007.

[12] M. Nikjoo S., A. Saber Tehrani, and P. Kumarawadu, "Secure routing in sensor networks," *Electrical and Computer Engineering, 2007. CCECE 2007. Canadian Conference on*, pp. 978–981, April 2007.

[13] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "Spins: Security protocols for sensor networks," in *Wireless Networks*, 2001, pp. 189–199.

[14] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: a link layer security architecture for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2004, pp. 162–175. [Online]. Available: http://dx.doi.org/10.1145/1031495.1031515

[15] A. D. Wood, L. Fang, J. A. Stankovic, and T. He, "Sigf: a family of configurable, secure routing protocols for wireless sensor networks," in *SASN '06: Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*. New York, NY, USA: ACM Press, 2006, pp. 35–48. [Online]. Available: http://dx.doi.org/10.1145/1180345.1180351

[16] J. Deng, R. Han, and S. Mishra, "Insens: Intrusion-tolerant routing for wireless sensor networks," *Computer Communications*, vol. 29, no. 2, pp. 216–230, January 2006. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2005.05.018

[17] B. Parno, M. Luk, E. Gaustad, and A. Perrig, "Secure sensor network routing: A clean-slate approach," in *Proceedings of the 2nd Conference on Future Networking Technologies (CoNEXT 2006)*, Dec. 2006.