CERIAS Tech Report 2007-04

SECURITY IN WIRELESS SENSOR NETWORKS - A LAYER BASED CLASSIFICATION

by Mohit Saxena

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

Security in Wireless Sensor Networks A Layer-based Classification

Mohit Saxena
Department of Computer Science
Purdue University, West Lafayette, IN 47907

Abstract— **With a widespread growth in the potential applications of Wireless Sensor Networks (WSN), the need for reliable security mechanisms for them has increased manifold. Security protocols in WSNs, unlike the traditional mechanisms, need special efforts and issues to be addressed. This is attributed to the inherent computational and communicational constraints in these tiny embedded system devices. Another reason which distinguishes them from traditional network security mechanisms, is their usage in extremely hostile and unattended environments. The sensitivity of the data sensed by these devices also pose ever-increasing challenges. We present a layer based classification of WSN security threats and defenses proposed in the literature, with special focus on physical, link and network layer issues.**

I. Introduction

Security is WSNs is an upcoming field, which is quite different from traditonal network security mechanisms. This is because of two major reasons. Firstly, there are severe constraints on these devices namely their minimal energy, computational and communicational capabilities. Secondly, there is an additional risk of physical attacks such as node capture and tampering. Hence the security issues in WSNs need to be addressed with highly light-weight and robust solutions. The different directions of ongoing research in WSNs are based on security challenges, including Key-Establishment Secrecy, Robustness to Denial-of-Service Attacks, Link Layer Security, secure routing, authentication and node capture. However, instead of addressing these issues on a per-attack basis, it is better to deal with them on a per-layer basis. Although in practical WSNs, such as those based on Berkeley Mica2 and Mica2dot Motes (Sensor Nodes), which use TinyOS platform, there does not exist a clear notion for demarcating between the various layers, but a clear understanding of WSN security certainly requires an abstract layer based classification. TinyOS specifications divide the communication stack into 3 major layers - Radio Stack (or the Physical Layer), the MAC layer (Data Link Layer which deals with issues such as power control, time scheduling and synchronization among the nodes) and the Application Layer (which is quite specific to the usage and deployment environment of the WSN).

II. Security Research in WSN: A layer based Survey

Before categorizing the ongoing research in WSN security on the basis of different layers, it is worth important to discuss the various key management schemes available in literature and their applicability to WSNs.

A. *Key Management in WSN*

1) Challenges to Key Management in WSN: Some of the major constraints which prevent from traditional key management and distribution schemes to be applicable to WSN can be enumerated as follows:

- Limited Processing and Memory
  A Berkeley Mica2 Mote has a tiny Atmega Microprocessor and 128 KBytes of programmable flash memory. Hence, running computationally intensive cryptographic algorithms over such tiny embedded system devices is infeasible. Public key cryptography is therefore almost ruled out for serving security in WSNs.

- Scalability
  A typical WSN may contain from hundreds to thousands of sensor nodes. So any protocol used for key management and distribution should be adaptable to such scales.

- Unique Communication Patterns
  Sensor nodes in a WSN possess a unique communication pattern. Therefore, security protocols and most important the key management should take care of these patterns. In a WSN, most of the communication links are established between a sensor node and the base station or between a sensor node and an aggregator node.

2) Key Management Schemes - A Survey: Following are the major key distribution and management schemes available in literature for traditional computer security.

- *Network Wide Shared Key*
  This is one of the simplest scheme, in which a single network wide symmetric key is used by every node. Here key distribution is almost absent and there is minimal communication overhead for key management. All the nodes use this key to establish secure communication links. However, this scheme is not resilient to the simplest single node compromise attack. An adversary can extract the network wide shared key by capturing a single node.

- *Master Key and Link Keys*

In this scheme every node in the network is preconfigured with a master key. Using this master key, every node fetches a set of link keys corresponding to its each communication link with other nodes. The links between the nodes are now secured using the keys from this set and the network master key is erased from all the nodes. This is resilient to a single node compromise attack. However addition of new nodes is not possible because once the master key is erased from all the nodes in the network, the link keys cannot be securely transmitted over the network.

- *Public Key Cryptography*
  Schemes such as Diffie Hellman propose a very good solution for key management and distribution in traditional wireless networks. However, the memory and processing constraints of these tiny devices rule out the possibility for using schemes based on public key cryptography.

- *Preconfigured Symmetric Keys*
  In this scheme, every node in the network is preconfigured with a set of link keys with which it will establish secure links with other nodes. However, this scheme is not scalable as every node has to store *n (n - 1)/2* keys, if *n* is the number of nodes in the network.

- *Bootstrapping Keys*
  This scheme allows an on-demand key generation for a secure connection established between the nodes. If a node has to communicate with any other node other than the base station, it will request for a link key from the base station. Base station will respond with a key which will be used for further communication. Again, this scheme suffers from single point of failure as base station has to maintain a database for the link keys.

3) Pairwise Random Key Pre-distribution Protocol: This is one of the recently developed key management and distribution protocol for WSNs [Du03]. In this scheme, the system has a large pool of symmetric keys. A random subset out of this pool is distributed to each sensor node. Now, two nodes can communicate with each other if they have a common shared key.

So there is an associated setup probability for every communication link. This probability is maximized using certain optimizations over the protocol. However, once a majority of nodes have been compromised with the adversary, the complete set of symmetric keys can be reconstructed.

Research in WSN key distribution and management is therefore focussed on two major issues. First is with providing sophisticated hardware support which can allow public key cryptographical algorithms to run efficiently on these tiny devices. Another focus is to develop better random key predistribution protocols which can maximize the associated link probabilities and also inhibit the reconstruction of the complete set of link keys from one of its subsets.

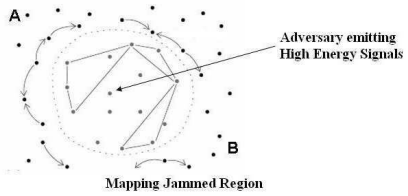

Fig. 1. Routing around the mapped JAM Region in a WSN.

B. *Physical Layer Security*

1) Jamming Attacks*:* This is one of the Denial of Service Attacks in which the adversary attempts to disrupt the operation of the network by broadcasting a high-energy signal. Hence, if the transmission is powerful enough, entire network's communication can be jammed. The defense mechanisms proposed in literature against this attack use spread-spectrum techniques for radio communication, so that the transmitter can communicate over a different cryptographically secure spectrum range.

Handling jamming over the MAC layer requires Admission Control Mechanisms so that requests intended to exhaust the power reserves of a node can be ignored. Network layer deals with it, by mapping the jammed area in the network and routing around the area, as can be observed in Fig. 1. The jammmed region can be mapped using a framework proposed by A. Wood [Wood03]. So that is much of detection instead of prevention. Other than jamming attacks there are radio interference attacks in which the adversary either produces large amounts of interference intermittently or persistently. Recently new techniques have been developed to handle this issue, through the use of symmetric key algorithms in which the disclosure of the keys is delayed by some time interval.

C. *MAC (Link) Layer Security*

1) Continuous Channel Access (Exhaustion): In this kind of attack which is also much prevalent in almost any Wireless Network, being it 802.11 - based communication or any other, a malicious node disrupts the Media Access Control protocol, by continuously requesting or transmitting over the channel. This eventually leads a starvation for other nodes in the network w.r.t channel access. This attack is usually done by transmitting a large number of RTS (Request to Send) packets over the media. So it leads into multiple collisions of the network packets, thereby the nodes draining out their power.

One of the countermeasures to such an attack is *Rate Limiting* as described in [Wood02]. Here, the network ignores excessive requests without sending expensive radio transmissions. This limit however cannot drop below the expected maximum data rate the network has to support. This limit is usually coded into the protocol during the design phase and requires additional logic also.

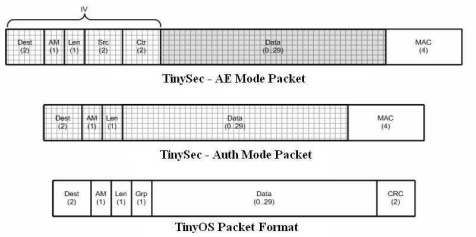

Fig. 2. TinySec: Security Modes [Karl04]

2) Collision: This is very much similar to the continuous channel attack discussed above. However in this attack, the adversary may only need to induce a collision in one octet of a transmission. A minute change in the data portion of the packet will result in a checksum change, hence requiring an expensive exponential back off in some MAC protocols.

Error Correcting Codes can be used to tolerate variable levels of corruptions in the messages at any layer. However these error correcting codes can only work upto a threshold of corruption and they themselves induce additional computational and communicational costs.

3) Unfairness: Repeated application of these *exhaustion* or *collision* based MAC layer attacks or an abusive use of cooperative MAC layer priority mechanisms, can lead into *unfairness*. This kind of attack is a partial DoS attack, but results in marginal performance degradation.

One major defensive measure against such attacks, is the usage of *small frames*, so that any individual node seizes the channel for a smaller duration only. However the adversary can still cause starvation by frequently requesting for channel, while others go on a random back off.

4) TinySec: This is a Link Layer Encryption Protocol for tiny devices such as sensor nodes, developed at University of California - Berkeley [Karl04]. This is very much similar to its IPSec counterpart frequently used in the computer networks of today. However, this works at MAC layer and provides authentication, access control and confidentiality.

It also works in 2 modes as is IpSec - TinySec - AE and Auth Modes as shown in Fig. 2. TinySec's performance evaluation has revealed an interesting phenomena that most of the performance overhead is attributed to the increase in packet size. In comparison, cryptographic computations have very less or no effect on the latency and throughput, because they usually overlap with the transmission.

New mechanisms to support physical and MAC layer level security in WSN are still needed to evolve. Some like *cryptographically secure spread spectrum* radios which can withstand physical layer jamming attacks need to be commercially available. Other defense mechanisms which are resilient to *node capture* attacks also need to be devised.

TinySec and *JAM* frameworks are the two only major defense mechanisms which provide considerable protection at link and physical layers. However these two schemes have their assumptions, eg. the JAM framework [Wood03] assumes single channel wireless communication such that no other communication channel is available for the legitimate nodes to communicate once the network has been jammed. Moreover, it also assumes that only small portions of the WSN are exposed to jamming. So even if multiple attackers perform simultaneous jamming attacks, in-network mapping is still possible as it is assumed that the whole network will never be jammed. Such assumptions may or may not be true depending upon the intensity of the attack. Hence, new schemes need to be devised which are much more resilient and effective.

D. Network Layer Security: Secure Routing

- *DoS Attacks over the Routing Protocols*

  Current routing protocols in WSNs or even in Wireless Ad hoc Networks are very susceptible to DoS attacks. The most simple among those is where the adversary injects malicious routing information into the network. This results in routing inconsistencies leading to high increase in end-to-end delays or even packet losses in the network. A good solution to this is based on Authentication over the Network Layer, which may guard against unauthenticated injections. However, authentication itself is not sufficient if the adversary replays the packets sent by legitimate nodes.

- *Node Capture Attacks*

  Routing protocols are highly susceptible to node capture attacks. It is observed and analyzed that even a single node capture is sufficient for an attacker to take over the entire network. Unlike traditional networks, where physical security can prevent such conditions, sensor networks belong to extremely hostile and unattended environments. A recent work has shown that the standard sensor nodes, such as MICA2 motes, can be compromised in less than one minute. Such exposure increases the possibility of attacker extracting secret cryptographical secrets, modifying their programming and even replacing them with malicious nodes under their control.

  Other than physical security, this problem requires algorithmic solutions. A good solution to this problem would definitely constitute a ground-breaking work in WSN. Networks resilient to such attacks typically use state replication across the network with majority voting to detect inconsistencies. A simple example of this approach is using multiple, independent paths for routing the packets and then detecting for inconsistencies among the received packets.

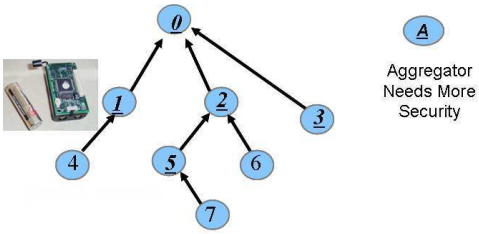

Fig. 3. Secure Group Management: Aggregators and Base Station

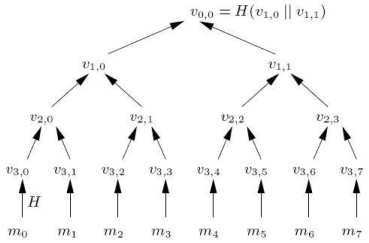

Fig. 4. Merkle Hash Tree: *Aggregate-Commit-Prove* Framework for Secure Information Aggregation [Przy03]

The second approach to model node-capture resilient networks, is by gathering multiple, redundant views of the environment and cross checking them for inconsistencies. When many data values or snapshots of the network are taken, a histogram is constructed and extreme outliers indicate malicious spoofed data.

Another important broadcast authentication scheme named μTesla. It provides a multi-level mechanism to withstand replay and denial of service attacks. This scheme induces low overhead, tolerance of message losses and is highly scalable.

A more formal look at the problem of secure routing is taken up again in section III, which covers the details of different network layer attacks and their countermeasures.

E. *High-level Security Mechanisms*

- *Secure Group Management*

  In-network processing of the raw data is performed in WSNs by dividing the network into small groups and analyzing the data aggregated at the group leaders. So the group leader has to authenticate the data it is receiving from other nodes in the group. This requires group key management. However, addition or deletion of nodes from the group leads to more problems. Consequently, secure protocols for group management are required. Moreover these protocols have to be necessarily efficient relative to time, energy, computation and communication. So the traditional group management approaches are ruled out and novel algorithms are needed. Special attention needs to be paid towards the security of aggregator nodes and the base station as shown in Fig. 3.

- *Intrusion Detection*

  The problem of intrusion detection is very important in the case of WSNs. Traditional approaches which do an *anomaly analysis* of the network at a few concentration points, are expensive in terms of network's memory and energy consumption. So there is a need for a decentralized intrusion detection which can analyze the network characteristics in a distributed fashion. Moreover, there also exists a need to understand the cooperative nature of adversary attacks.

- *Secure Information Aggregation*

  Usually, a WSN consists of a large number of sensor nodes which are deployed in some area distant from the *home server*. These sensor nodes perform measurements and route the information towards the base station. However in order to save the communication bandwidth, these readings are aggregated at intermediate points in the network which are called as aggregators. Some sensor networks have a single aggregator, which is usually the base station itself and others such as [Madden02] have multiple aggregators where each non-leaf node is an aggregator, as also shown in Fig. 3.

  In this setting, there are two major attacks over the information being aggregated, as described in [Przy03]. First is the *stealthy attacks*, in which the attacker's goal is to make the home server accept false aggregation results, which are very much different from the actual results determined by the measured values. Moreover, the attacker also wishes that the homing server is not able to detect these changes. So he does not launch a *denial of service* attack by not responding with the aggregated values at all.

  Stealthy Attacks are much more difficult to detect. [Przy03] tackle this problem by using an *aggregate-commit-prove* strategy which is based on a variant of Merkle Hash Tree based information aggregation and authentication, as shown in Fig. 4. The aggregator constructs a merkle hash tree over the measured values as the leaf nodes.

F. *DoS Attacks in a WSN*

Most of *DoS Attacks* in WSNs have been already discussed under the categories of Physical Layer (Jamming Attacks),

Link Layer (Exhaustion, Unfairness and Collision) and Network Layer (Homing and Misdirection, Black Hole Attacks). However two major *DoS Attacks* have not been still discussed. These actually fall under the category of *Transport Layer* Security.

- *Flooding Attacks*

  Flooding Attacks produces severe resource constraints for legitimate nodes. Two major countermeasures have been devised to tackle such kind of attacks.

  Limiting the number of connections which a node can make prevents complete resource exhaustion. But this would interfere with all the other processes at the victim and will prevent the legitimate clients to connect to the victim.

  Another solution is based on an idea of *Client Puzzles* [Aura00]. Considering the server-client model, the server creates and distributes puzzles to the potential clients wishing to connect to it. So, in this manner an adversary must therefore commit for more computational resources per unit time to flood the server with valid connections. Under heavy load, the server could increase the scale of puzzles to require even more work by the potential clients.

- *Desynchronization Attacks*

  In this attack, the adversary repeatedly forges messages to one or both end points which request transmission of missed frames. Hence, these messages are again transmitted and if the adversary maintain a proper timing, it can prevent the end points from exchanging any useful information. This will cause a considerable drainage of energy of legitimate nodes in the network in an end less synchronization-recovery protocol.

  One solution to counter this attack requires *authentication* for all packets exchanged, including all control fields in the transport header which represent information for missed frames and sequence numbers. The end points can now detect a *Desynchronization Attack* as the adversary cannot provide a modified packet with unmodified and authentic header information.

III. The Problem of Secure Routing in WSN: A Formal Look

The most important security issue in WSNs is its inherent security limitations relative to routing. Before discussing the various possible attacks and their countermeasures in a WSN, a brief formalization of general Wireless Sensor Network assumptions, trust requirements and security goals to achieve, is very much needed.

A. *Network Assumptions and Trust Requirements*

Like all other wireless networks, three major security vulnerabilities exist in WSN too, due to the insecure radio links. Firstly, the attackers can eavesdrop on the radio transmissions. Second, they can inject malicious bits over the channel. Thirdly, they can replay the previously heard packets. It is quite reasonable to make the assumption that if the defender can deploy many sensor nodes, so can the adversary. We also assume that if the attacker can compromise a node, he can extract all the key material, code and data stored on that node and can even replace or modify it.

The major trust requirements for the issue of security in WSN are related to the behavior of the *aggregation points* in the network - whether it is the *Base Station(s)* which acts like a bridge between the sensor network and the outside world, or it is some aggregation node in the WSN itself. The security and trustworthiness of these nodes in a WSN are of utmost importance. Hence, we assume that nodes can rely on routing or any other information from the base stations and the data aggregation points in the network. However in some situations, aggregation points may not be considered trustworthy.

B. *Threat Models*

As discussed in [Karl03], security threats in a WSN can be divided into two major categories - *Mote Class Attackers* and the *Laptop Class Attackers*. Another division can be based on the basis of *Insider Attacks* and *Outsider Attacks*.

Mote Class Attackers possess the capabilities similar to a sensor node or a few sensor nodes. However, a laptop class attacker may have an access to more powerful devices, such as laptops. Hence, they possess higher processing power, better radio transmitters and more sensitive antennas.

Outsider attacks are more common in WSN, where the attacker had no special accesses to the system in the beginning. However insider attacks can be much more dangerous than the outsider ones. Insider attacks can be mounted from compromised nodes running malicious code and are much difficult to detect. These can also be mounted from motes whose key material, code or the data has been stolen by the adversary.

C. *Attacks on Routing in WSN*

Most of the network layer attacks can be categorized into the following classes of attacks.

- *Selective Forwarding*

  WSNs are usually multi-hop networks and hence based on the assumption that the participating nodes will forward the messages faithfully. Malicious or attacking nodes can however refuse to route certain messages and

drop them. If they drop all the packets through them, then it is called a *Black Hole Attack*. However if they selectively forward the packets, then it is called *selective forwarding*. These attacks are typically most effective when the attacker is explicitly included on the path of a data flow. However, an attacker may also be able to jam the network by simply causing collisions of packets of interest.

To include himself on the path of the data flow, the adversary can use two major strategies which correspond to the *Sink Hole Attacks* and the *Sybil Attacks*.

- *Sink Hole Attacks*

  Sink Hole attacks are based on the idea that the adversary can lure some or most of the traffic in a certain region of the network, by spoofing or replaying an advertisement for a high quality link to the base station. Some routing protocols try to verify the bidirectional reliability of a route with end to end acknowledgements which contain information regarding the reliability or latency information. When we consider the laptop-class adversaries with a powerful transmitter which can actually provide a high quality link between a node and the base station, then the adversary can easily dupe the other nodes.

  The adversary creates a large *sphere of influence*, which will attract all traffic destined for the base station from nodes which may be several hops away from the compromised node.

- *Sybil Attack*

  In this attack, a single node presents multiple identities to all other nodes in the WSN. This may mislead other nodes, and hence routes believed to be disjoint w.r.t. node can have the same adversary node. This attack poses a major concern for *Geographical Routing Algorithms* which require the location of a node to efficiently route the message. If the same adversary node shows as to be at more than one place, then most of the geographically addressed packets will get routed to this adversary node, leading to Selective Forwarding.

- *Wormhole Attacks*

  An adversary can tunnel messages received in one part of the network over a low latency link and replay them in another part of the network. This is usually done with the coordination of two adversary nodes, where the nodes try to understate their distance from each other, by broadcasting packets along an out-of-bound channel available only to the attacker.

  As shown in Fig. 5, this is an example of Worm Hole Attack on TinyOS Beaconing based routing protocol, in which a tree is created with the base station as the root and all other nodes communicate through their parents to the base station. Here as seen in the figure, two adversaries mislead the other nodes in the network by using a powerful out-of-bound channel between them. Hence a node which is much far (several hops) to the base station, gets the view that if it uses the adversaries along the routing path, then it will be able to reach the base station in just a few hops.

*D. **Secure Routing: Countermeasures***

Now as we have discussed some of the major attacks on WSN routing protocols, we can proceed to know the techniques available in literature to counter them.

- *Selective Forwarding*

  A compromised node can be forced to be always at a data flow path and hence launch a selective forwarding attack as discussed earlier. Two major countermeasures exist for such attacks.

  Multipath routing can be used in combination with random selection of paths to destination, to counter selective forwarding attacks. Packets routed over n disjoint paths which have no common node, are completely protected against selective forwarding attacks involving atmost n compromised nodes and are still probabilistically safe for over n compromised nodes.

  As finding completely disjoint paths is relatively hard, *braided paths* can be used which represent paths which have no common link or which do not have two consecutive common nodes. These provide probabilistic protection against selective forwarding attacks. This scheme when integrated with dynamic choice for next hop at every intermediate node for a packet will further reduce the chances of a selective forwarding attack.

- *Sink Hole and Worm Hole Attacks*

  Sinkhole and Wormhole Attacks are very difficult to detect esp. in WSNs which use a routing protocol in which routes are based on advertised information such as remaining energy or an estimate of end-to-end reliability or minimum hop count to base station.

  A class of routing protocols which is resilient to such kind of attacks is based on the geographical location of the nodes, eg. GPSR (Greedy Perimeter Stateless Routing) [Karp00] and GEAR (Geographic and Energy Aware Routing) [Yu01] protocols. These *geographic*

routing protocols are able to figure out the actual location of the adversary nodes and do not rely on the misleading advertisements by them. Hence the traffic is routed to the base station along a path which is always geographically shortest.

Another technique proposed in literature [Hu02], is able to tackle wormhole attacks but requires very tight time synchronization among the nodes which is infeasible in practical environments.

- *Sybil Attacks*

  Identity verification is imperative to detect and mitigate sybil attacks in a WSN. In a traditional network, identity verification can be done using a single shared symmetric key and public key algorithms. However this is infeasible for WSN due to computational constraints.

  One of the major countermeasures to Sybil Attack [Douc00]is by using a unique shared symmetric key for each node with the base station. In this way, two nodes can use a *Needham - Schroeder* like protocol to verify each other's identity and then setup a shared key to implement an authenticated, secure and encrypted link between them. In this scheme, when a node is compromised, it will be then restricted to communicate only with its verified neighbors. In this way, the compromised node is only present on the data flow path of a limited number of nodes, thereby mitigating the affects of Sybil Attack.

  So, in order to ensure that no insider node in the network is able to verify and authenticate itself with every other node in the network, the base station can always limit the number of neighbors with which a node is allowed to have shared keys. In this way, the affects of Sybil Attack can be mitigated.

- *Authenticated Broadcast and Flooding*

  Hello flood attacks and acknowledgement spoofing also form a major category of attacks on routing protocols in WSNs. Authentication is the key solution to such attacks. Authenticated broadcast protocols such as μTesla [Perrig01] are examples of efficient protocols which although being based on symmetric key cryptography requires minimal packet overheads. μTesla achieves the asymmetry necessary for authenticated broadcast and flooding by using *delayed key disclosure*. Replay is prevented because messages authenticated with previously disclosed keys are discarded. μTesla also has the advantage that it does not require tight time synchronization between the nodes in a WSN.

  The downsides of flooding, namely high messaging and energy costs, as well as potential losses caused by collisions, are taken care in recent techniques such as SPIN [Kulik02] and other gossiping algorithms. These protocols reduce messaging and collision overheads and still provide robust probabilistic dissemination of messages to every node in the network.

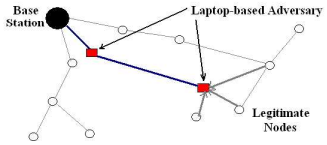

Fig. 5. Wormhole Attack on a TinyOS Beaconing based routing protocol in a WSN.

E. *Summary*

Many techniques such as Link Layer encryption and authentication, multipath routing, identity verification and authenticated broadcast seem to be good solution for security in a WSN. However, attacks such as *Sinkhole* and *Wormholes* pose lots of challenges to secure routing protocol design. Geographic Routing Protocols is one example of routing protocols which are able to withstand most of the WSN routing based attacks, as the legitimate nodes are able to estimate the location of the adversary nodes. Hence attacks such as Sybil are rendered ineffective. Effective countermeasures are still lacking against these attacks, which can be applied after the design of these routing protocols has completed. So there exist a severe need to design such routing protocols in which these attacks are ineffective.

IV. Potential Ideas and Critique

A. *Node Address Hopping*

In sensor networks, every mote (node) is assigned a node id within a range say [1,n]. Now, let us assume an attacker inserts a malicious node (intrusion) in the network which launches a Denial of Service Attack on the routing protocol by injecting misleading routing packets in the network destined for nodes with addresses in this range. However, if we do a periodic hopping of the Range of Node Ids, for example we use a Memory-Less Random Distribution for generating a new set of Node Ids for the n nodes in our network. Then it could become quite difficult for the malicious node to introduce routing inconsistencies in the network. This mechanism is used in real networks, but is novel to the area of WSNs.

The major advantage is that this is a simple but very efficient solution to both the problems of Network Intrusion and Denial of Service Attacks on Routing Protocols. This has very low overheads as we do not need the routing tables to be resent and updated, as every node which is not malicious can easily predict the new node ids of its neighbours based

on the generating function. This is also scalable as there are no constraints over the available node id space in the present WSNs. However there is a minor computational overhead of this scheme due to the local updatations at each node. We assume that time synchronization is already present in the network, which is usually the case.

B. *Periodic Authentication*

Network Layout: We have a sensor network of n nodes where the nodes are assigned Node Ids 1 to n. We have a key-establishment and trust setup in the network which is either based on a single network-wide shared symmetric key or a large pool of symmetric keys in a random-key pre-distribution protocol.

Attack: We suppose there is a network intrusion attack, where the attacker inserts a Malicious Node in the network with a node id in the range [1,n]. We also assume that the secret information which was present in the network prior to intrusion has also been compromised with the intruder.

This secret information can be a single network-wide shared symmetric key in the most trivial trust protocol. Or it can be even a major subset of randomly distributed symmetric keys in case of the random - key pre-distribution protocol. So our basic problem now converges to how to detect the Malicious Node, even if it has got hold of this secret information. All the authentication mechanisms which relied on this secret information will fail, whether it is the TinySec (MAC layer level authentication) or it is a Network Layer Level Encryption of Data Payload.

Possible Solution: Our major problem is to detect the Malicious Node, once it has entered the network and got hold of the secret information (say S-old). So when, one level of authentication has failed, we need a mechanism to detect this. A possible solution can be to have another layer of a "periodic" authentication scheme (say P-Auth), which gets invoked after certain intervals of time and the secret information (say SP) for this is generated on the fly, using a mechanism which we assume is not compromised with the intruder. This mechanism can be as simple as using S-old in reversed order as SP. SP can be even generated on the fly as a function f(P, S-old) where P is the number of periodic intervals elapsed since the network came into being.
There are two major reasons for the correctness of this solution:

- We are invoking this P-Auth protocol periodically and its secret information is generated on the fly. Hence the malicious node which we suppose has got hold of the older secret information S-old, will still not be able to pass this P-Auth test as we assume that an intruder can get hold of the embedded secret keys and information, but not the mechanism.

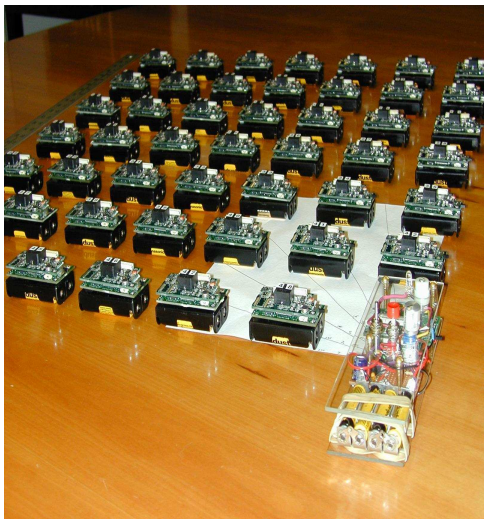

Fig. 6. Berkeley Mica2 Sensor Motes in Action

- Moreover as this mechanism is invoked periodically among all the network nodes except the malicious node, we are sure that at the next periodic instance, we can be sure to detect the malicious node in the system. We can even the reduce the overheads by only running this protocol over a random fraction of selected nodes in the network.

V. Conclusion

The challenging constraints and demanding hostile deployment environments, make network security in WSNs more challenging as compared to the traditional networks. Many problems have their solutions using asymmetric key or computationally intensive protocols, but they do not suite the requirements of these tiny power and memory constrained devices. Hence there still exist a major need to evolve novel security techniques for WSNs which are much more efficient and take care of the constraints of these tiny devices. Through this study, we have explored the current research potential and its future directions in this field of Security in WSNs, along with a critical analysis of the present security mechanisms.

References

[Perr04] Adrian Perrig, John Stankovic, and David Wagner. Security in Wireless Sensor Networks. In *Communications of the ACM Vol. 47, No. 6*, 2004.

[Walt06] John Paul Walters and Zhengqiang Liang. Wireless Sensor Network Security. In *Security in Distributed, Grid and Pervasive Computing*, 2006.

[Karl03] Chris Karlof and David Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*. 2003.

[Karl04] Chris Karlof, Naveen Sastry and David Wagner. TinySec: Link Layer Encryption for Tiny Devices. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004). November 2004.*

[Hu02] Y.-C. Hu, Adrian Perrig and D.B. Johnson Wormhole Detection Wireless Ad Hoc Networks *Technical report, Rice University Department of Computer Science, June 2002.*

[Wood02] Anthony D. Wood and John A. Stankovic Denial of Service in Sensor Networks *IEEE Comput. (Oct. 2002), 54 - 62.*

[Przy03] B. Przydatek, D. Song, A. Perrig Secure Information Aggregation in Sensor Networks *ACM SenSys 2003 (Conference on Embedded Networked Sensor Systems).*

[Du03] W. Du, J. Deng, Y. S. Han, P. K. Varshney A Pairwise Key Predistribution Scheme for Wireless Sensor Networks. *10th ACM Conference on Computer and Communications Security (CCS), Washington DC, October 27-31, 2003.*

[Wood03] A. Wood, J. Stankovic, and S. Son JAM: A mapping service for jammed regions in sensor networks. *IEEE Real-Time Systems Symposium, Cancum, Mexico, Dec. 3-5, 2003.*

[Madden02] S. Madden, M. Franklin, J. Hellerstein, and W. Hong TAG: a Tiny Aggregation service for ad-hoc sensor networks. *Fifth Annual Symposium on Operating Systems Design and Implementation (OSDI), December 2002.*

[Perrig01] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar SPINS: security protocols for sensor networks. *Mobile Network and Computing, 2001*

[Kulik02] J. Kulik, W. R. Heinzelman, and H. Balakrishnan Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks. *Wireless Networks, vol. 8, no. 2-3, pp. 169-185, 2002.*

[Karp00] B. Karp and H. T. Kung GPSR: greedy perimeter stateless routing for wireless networks. *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking.*

[Douc00] J. R. Douceur The Sybil Attack. *1st International Workshop on Peer-to-Peer Systems (IPTPS'02), March 2002.*

[Yu01] Y. Yu, R. Govindan and D. Estrin Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. *UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023, May 2001.*

[Aura00] T. Aura, P. Nikander, and J. Leiwo DOS-Resistant Authentication with Client Puzzles *Proc. Security Protocols Workshop 2000, Springer-Verlag, New York, 2000, pp. 170-177.*