Using Routing Data for Information Authentication in Sensor Networks

Vijay Bhuse, Ajay Gupta, Mark Terwilliger, Zijiang Yang, Zill-E-Huma Kamal
Computer Science Department, Western Michigan University
{vsbhuse, gupta, mgterwil, zijiang, zkamal}@cs.wmich.edu

Abstract- **Wireless sensor networks (WSNs) are envisioned to be used in a wide variety of applications for deep monitoring of the surroundings. Their deployment in hostile environments, however, faces many security challenges. Wireless communication is inherently broadcast and insecure. If WSNs are deployed in unregulated environments, an adversary can easily tamper with them or compromise some nodes. Sensor nodes are severely resource constrained in terms of power, memory and processing abilities. Achieving security for these networks is thus a challenging task. Sensor networks typically follow specific communication patterns. Effective security primitives can thus be provided by taking the application specific nature of WSNs into consideration. In this paper we propose a simple, lightweight and scalable protocol, Information Authentication in Sensor Networks (IASN) that can aid in providing information (high level data) authentication. IASN is able to detect and filter a significant number of forged packets at low cost.**

I. INTRODUCTION

WSNs are being used or envisioned to be used in a wide variety of applications such as environmental monitoring, fire alarms and military. Researchers have so far mainly focused on making sensor networks feasible, useful, robust and reliable. Only a few researchers have considered the secure communication aspects in WSNs (see for example [BG03, KS03, PS01, PS04]). An adversary can attack or confuse the WSN in a variety of ways. If sensors are deployed for sensing hazardous leak at a chemical plant, an adversary can suppress an alarm from ringing when there is a leak. If sensors are deployed for monitoring a railway crossing, an adversary can cause accidents. One can easily imagine numerous examples of the use of sensor networks where an adversary can create havoc.

The resource-constrained nature of WSNs poses great challenge for secure communication. Physical and link layer security mechanisms are not enough. Routing protocols should also contribute in making communication secure and hence they must be designed with security in mind [KW03]. Security primitives like signatures, encryption and one-way functions are computationally intensive. Sensor networks are energy constrained so implementing these traditional security primitives on the resource-constrained nodes does not seem very practical.

The resource-constrained nature of WSNs motivates us to develop lightweight protocols that use already available system information efficiently. One such possibility is to use routing path details to provide additional security by detecting and filtering forged packets. We develop this idea in the paper by proposing a simple, lightweight and scalable protocol, Information Authentication in Sensor Networks (IASN), which can aid in providing information (high level data) authentication.

Sensor networks typically follow some specific communication patterns (e.g. sense temperature of an area and send results back to a sink node periodically) and the end user is mainly interested in the high-level data. Generally a sink node sends a query and nodes do some collaboration to find an answer, which is sent back to the sink. Aggregators play an important role in reducing communication by computing high-level data as an answer to a query. The sensor nodes may be deployed in thousands and the collective result is more important than the data from individual sensors. Security can thus be provided taking into consideration the specific communication patterns and importance of interest in high-level data.

The high-level data (information) is generated by a group of collaborating nodes. This motivates consideration of the following type of authentication: *Enforcing validity of high-level data is more important than enforcing validity of the source.* If information is authenticated, source authentication is implicitly achieved. In this paper our focus is thus on attempting to provide information authentication for WSNs by detecting and dropping significant number of forged packets at a minimal cost. The basic idea behind our proposed IASN protocol is to maintain a data-source table at the nodes running IASN. This data-source table keeps track of predecessors from which a particular type of data is expected and is derived from the existing routing data. Furthermore, it is dynamically updated as the route changes occur. The overheads of IASN are minimal as it uses existing routing information. The novelty of the proposed IASN protocol is in its simplicity.

For a good survey of related work on security in sensor networks we refer the reader to [PS04, WS02]. To the best of our knowledge, currently there are no proposed protocols to detect forged packets in WSNs. We try to address this problem in the paper. The rest of the paper is organized as follows. In Section 2 we give the design of IASN protocol. Discussion and simulation results are presented in Sections 3 and 4 followed by conclusions and future work.

II. IASN PROTOCOL

Let us first consider Directed Diffusion [IG00] and Data Centric Storage (DCS) [RE02] techniques that are proposed for WSNs. We then show how these techniques can be easily extended to provide information authentication, which forms the basis of the IASN protocol design.

Directed diffusion is data centric. Data is identified by a set of attribute-value pairs. A sink (node interested in

some kind of data) generates an interest (a query) and floods it in the network. While an interest is flooded, the nodes that receive the interest establish gradients. Gradients are the next hop direction of other nodes with matching interest. Whenever a source node with matching interest is found, it sends the data back to all the neighbors that have matching gradients. The reverse path is found using the neighbor information and the gradient (which depends on how efficiently data is received) towards neighbor. If this path is broken it is reinvented. When the initial data reaches a sink, the sink reinforces its neighbor, which in turn reinforces its neighbor. In this way an entire path from sink to source is reinforced which is then used for sending data periodically. Data can be cached at intermediate nodes. Data aggregation also takes place (to minimize communication) near the place where data is sensed.

An adversary can quickly learn the communication pattern in the directed diffusion by eavesdropping on the periodic packets sent using the same path. Once the adversary finds out the path from source to sink, it can infuse forged packets on the path destined to sink with forged data values. We can detect infusion of such forged packets by keeping track of predecessors on the routing path for each type of data at the intermediate routing nodes. For example, if directed diffusion establishes a path from source node C to sink node A via node B for temperature data, then node A expects temperature data from node B and node B expects temperature data from node C. If nodes A and B receive the temperature data from any other nodes except nodes B and C, respectively, then they can detect and filter (drop) forged packets. This idea, although simplistic, is the main idea behind the lightweight IASN protocol.

Data centric storage (DCS) stores different types of data at different locations [RE02] by using a Geographic hash table (GHT) for storing data. A GHT is a mapping from data to location. The nodes are assumed to be location aware. They store their neighbor list and locations of neighbors. The main advantage of DCS is that nodes know the location at which the required data is stored. Therefore they can directly send data request towards the direction of the source node's location. It is again easy to see that an adversary can learn the communication pattern and infuse forged packets by making it look like that the data is coming from the correct (source) location. This type of forging can be easily detected and forged packets filtered by keeping a list of predecessors on the routing path for each type of data at the intermediate nodes.

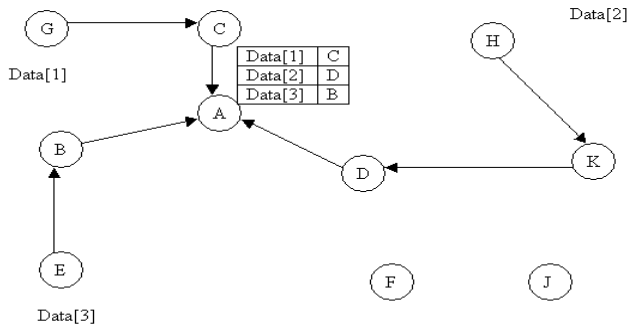

Figure 1: DCS for a WSN and data-source table of node A.

Consider the scenario in Figure 1 that uses DCS for a WSN. Suppose nodes G, H, E store Data[1], Data[2], and Data[3] respectively. Node A has established paths to get data from these nodes. Node A gets Data[1] from node C, Data[2] from node D and Data[3] from B. Node A can maintain a data-source table, which contains information about the type of data and the predecessor from which it is expected. Now suppose there is an adversary F, which sends a forged packet containing Data[1] to node A (see Figure 2). If node A gets this packet from node B or D then it can detect that the packet is forged. If node A keeps information about route updates and keeps the data-source table consistent with the current routing paths then it can very easily detect packet spoofing. Note that adversary F can spoof a Data[2] packet via node D. However, if node D also maintains its data-source table and runs an IASN protocol, then this forged packet can also be detected. It is clear that there is a tradeoff between the number of forged packets detected and the number of nodes running an IASN protocol.

Suppose directed diffusion is used in the scenario of Figure 1 where node A is a sink which has established paths with node C for Data[1], with node D for Data[2] and with node B for Data[3]. In this case node A can again keep track of route updates and the information about the type of data and the nodes from which it arrives. Designers can thus use the system information to make the network less vulnerable. The above examples show that the system can be made robust and sustainable to packet forging attacks by a simple design, which is the essence of the lightweight IASN protocol.

Let T be the number of different types of data handled by an application and Data[i] indicate the type of data for $1 \leq i \leq T$. Every node running IASN maintains a data-source table M indicating the list of neighbors that may forward Data[i] to that node. IASN protocol then detects and drops forged packets by comparing incoming packets against the entries in the data-source table. Efficient representation of M should be used to optimize the performance of IASN with respect to time and space requirements. Figure 3 gives an example of data-source tables at various nodes in the WSN.

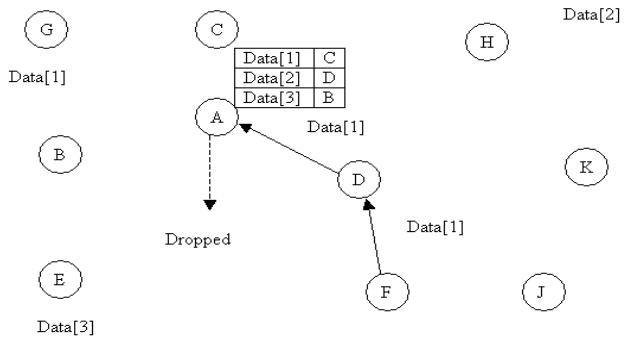

Figure 2: Detecting a forged packet from adversary F at node A.

In Figure 3, nodes G, H, E, F and J have empty data-source tables, as they are not receiving any data. Whereas A, B, C, K and D have some entries in their tables. Figure 4 shows an example where adversary J is detected at node D. In this example all the forged packets will be detected and dropped if

all the intermediate nodes run the IASN protocol and nodes do not masquerade. (Detecting masquerading without an explicit node authentication mechanism, such as one-way function, secret identity or signature is a challenge in wireless networks. Furthermore, our goal in this paper is to show efficient use of existing system information to make the communication more secure at a low cost.)

In the previous examples a node receives data of a certain type from a single neighbor. The data-source table can be easily extended to accommodate the situations where (i) multiple neighbors are allowed to forward the same type of data, or (ii) multiple types of data are forwarded by a neighbor. Efficient data structure to represent the data-source table can also be designed accordingly. It is easy to see that the storage overhead in IASN is similar to a DSDV[PB94]

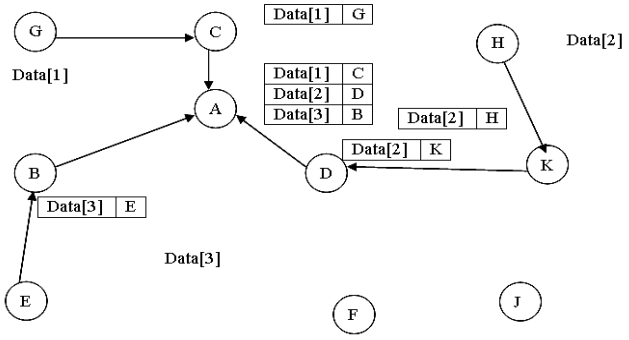

Figure 3: Data-source tables at various WSN nodes.

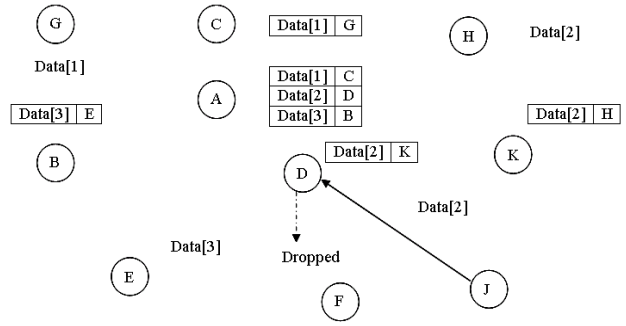

Figure 4: Forged packet containing Data[2] from adversary J is dropped at node D.

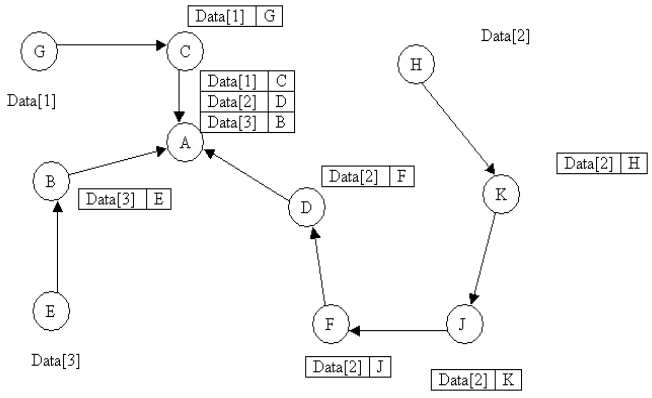

Figure 5: Data-source tables at various WSN nodes after route change.

type of ad hoc routing protocols. Furthermore, data-source tables can be easily derived and maintained from the underlying data dissemination mechanisms or the routing protocols. The forwarding-tables / next-hop information of the routing protocols can be used to build the data-source tables.

Furthermore route-update messages can be used to maintain the data-source tables in concurrence with the routing path changes. Figure 5 shows the updated data-source tables after a H→D path changes from Figure 3. IASN is thus lightweight because it uses existing routing information.

III. DISCUSSION

Note that the design of any protocol that provides information authentication must consider the following requirements: It must be independent of the data dissemination mechanism. It must adapt to the updates in the established routing paths. The protocol itself has to be secure. It can be deployed incrementally and function in partial deployments as well. Finally, it must be lightweight and energy aware to be practical for the resource constrained WSNs. IASN meets these requirements.

We assumed that node locations are fixed for the lifetime of the sensor network. We also assumed that the adversary comes into play after deployment and does not interfere with initial path establishments. For the IASN protocol to work the following conditions must also be satisfied,

1. Data from source to sink should follow the route in accordance with the underlying routing protocol.
2. Data source table (which is used by IASN) should be updated after any routing changes.

Any node in a WSN, a base station, a sink or an aggregator, can use IASN. If only a sink uses this protocol, then a clever attacker can forge a packet by sending the data through the neighbor, which is supposed to forward it. If all of the nodes use this protocol then the forged packets can certainly be detected. It is possible to run this protocol on fewer nodes. In that case it will show better results than running IASN only on the sink nodes.

IASN has an overhead of maintaining and updating data-source tables. Stale entries in the data-source table can give inconsistent results. The table has to be updated whenever there is any change in the established paths through which nodes receive data. By incorporating IASN with the underlying routing protocols, one can eliminate any additional communication messages due to IASN and minimize the overhead in terms of maintenance of the data-source tables. The storage overhead is proportional to the connectivity of the network and the number of data types handled.

As we are not using any cryptographic techniques (like encryption, one way functions, etc) handling masquerading is beyond the scope of this paper. Depending upon other security primitives used (like encryption, one way functions) one may or may not use IASN. For example if effective authentication mechanisms like signature or MAC (which will incur considerable costs) are used then IASN is redundant. We believe that until the time these costly alternatives are feasible, IASN can provide a simple and efficient way for information authentication.

IV. SIMULATION

We simulated the above protocol using ns2 [NS2]. For our simulations we considered an area whose boundary is defined as 100m x 100m. We tested IASN with two routing protocols DSDV [PB94] and DSR [P97]. For each routing protocol we considered two types of topologies: fixed and random to simulate regular versus irregular placements of sensor nodes. This results in four scenarios. In fixed topology 100 nodes are arranged in a 10 x 10 grid and are uniformly distributed over the area. Whereas in random topology, we placed the nodes randomly in the 100m x 100m area. For all four scenarios an adversary is in one corner and a node under attack is in a diagonally opposite corner. We considered four types of data. An adversary is trying to inject packets of some type of data destined to the node, which is considered to be under attack. Then we varied the number of nodes that are running the IASN protocol. The nodes that run the IASN protocol are selected randomly. This experiment was repeated for 1 to 99 nodes that run the IASN protocol for all four scenarios.

The experimental results show that the number of packets that are detected increases as the number of nodes that run the IASN protocol increased. This was observed for both the routing protocols DSDV and DSR. Depending upon the communication pattern, designers can strategically select nodes to run the IASN protocol. They can create a huddle around a node that is most vulnerable. Selection of nodes across the network for optimal performance of the IASN protocol is obviously an NP-Complete problem. Factors such as number of types of data, communication pattern (represented by the data-source tables), topology and strategic location will affect the selection of a node to run the IASN protocol.

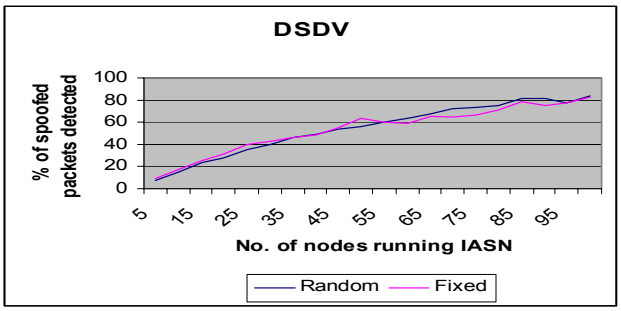

Figure 5: IASN with DSDV.

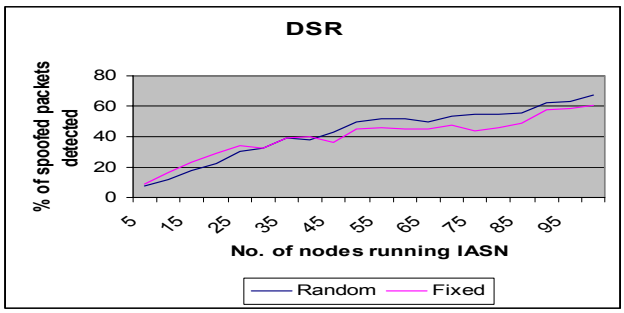

Figure 6: IASN with DSR.

V. CONCLUSIONS

We have proposed a lightweight and scalable protocol IASN, which can detect and filter a significant number of forged packets in sensor networks that use attribute-based naming for data dissemination. A base station or any other node in a network can use it. It will show better results with incremental deployment. Selection of a minimum number of nodes to run IASN protocol to achieve desirable performance is a challenge and part of our future work. IASN can also be used with other security primitives (like encryption). We believe that until the time costly authentication alternatives, such as signatures, are feasible for WSNs, IASN can provide a simple and efficient way for information authentication. We have shown that information authentication can be provided for resource-constrained wireless devices by using system information. Our future work also involves exploiting other system information, such as physical and link-layer protocols data to handle various other types of security threats such as masquerading.

ACKNOWLEDGEMENT

Research is supported in part by the National Science Foundation, under grants ACI-0000442, ACI-0203776, and MRI- 0215356, by the Department of Education grant R215K020362, and a Congressional Award, administered by the US Department of Education, Fund for the Improvement of Education. The authors would also like to acknowledge Western Michigan University for its support and contributions to the Wireless Sensor Network Research Laboratory, Computational Science Center and Information Technology and Image Analysis (ITIA) Center. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies or institutions.

REFERENCES

[BG03] V. Bhuse, A. Gupta and R. Pidva, “A Distributed Approach to Security in Sensornets”, In Proc of the IEEE VTC October 2003.

[IG00] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. Technical Report 00-732, University of Southern California, March 2000.

[KS03] C. Karlof, N. Sastry and D. Wagner , TinySec: Link Layer Security for Tiny Devices [Online], Available: www.cs.berkeley.edu/~nks/tinysec/

[KW03]Chris Karlof and David Wagner ,"Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", citeseer.ist.psu.edu/572017.html

[NS2] http://www mash.cs.berkeley.edu/ns.

[P97] C. E. Perkins, "Ad-hoc on-demand distance vector routing," in MILCOM '97 panel on Ad Hoc Networks, Nov. 1997.

[PB94] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In Proc. of the ACM SIGCOMM, October 1994.

[PS01] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. D. Tygar. SPINS: Security protocols for sensor networks. In Proceedings of MOBICOM, 2001

[PS04] A. Perrig, J. Stankovic and D. Wagner, “Security in Wireless Sensor Networks.” CACM, vol. 47, number 6, pages 53-57, June 2004.

[RE02] S. Ratnasamy, D. Estrin, R. Govindan, B. Karp, S. Shenker, L. Yin and F. Yu, Data-centric storage in Sensornets, Submitted for review. February 1st, 2002.

[WS02]http://www.ccs.neu.edu/home/zhufeng/security_manet.html