A Secure Routing Protocol for SensorNet

Daniel T. Fokum and Gary J. Minden
The Information & Telecommunication Technology Center (ITTC)
University of Kansas
2335 Irving Hill Road
Lawrence, Kansas 66046
USA
{fokumdt, gminden}@ittc.ku.edu

ABSTRACT
The secure delivery of data between a source node and a sink node is an open problem in wireless sensor networks. We review the security model used for The University of Kansas's SensorNet project, and discuss how it might be improved by using a secure routing protocol. In this paper we extend work done in developing multipath extensions to AODV. Unlike previous proposals on multipath extensions to AODV, the scheme proposed in this paper allows each node to use multiple disjoint paths concurrently to reach another node. Each of the paths is validated by using zero-knowledge proofs. Under the new scheme data is protected by traditional cryptography as well as using separate paths for key and data delivery. This increases the resiliency of the network as well as the security of the data.

KEY WORDS
Protocols for Sensor Networks, Security, Encryption

1. Introduction

Sensor networks are one of the emerging applications of computers. Sensor networks typically consist of a set of small resource-constrained computers, called sensor nodes that collect data from their environments and then collaborate to transmit that data on to a sink node (base station). In general a wireless sensor node (WSN) would consist of a sensing device e.g. an electronic nose, a temperature sensor or a motion detector etc, a small microprocessor, a radio and a limited energy source. Base stations, on the other hand will generally have radios, but will have available more computing resources and a larger energy source. The base stations will generally aggregate information from the nodes and then pass them on to other computers for presentation [1].

This work was supported in part by a grant from Oak Ridge National Laboratories (Grant FED 41420).

Since sensor networks are based on resource-limited computers that use wireless communication, sensor networks provide security challenges. Consequently any attempt to secure a sensor network must balance the energy consumption and computation overhead of the scheme with the security provided.

In this paper we review security in the ITTC's SensorNet project, with emphasis on routing between nodes. The rest of this paper is laid out as follows. In section 2 we provide a review of previous research on sensor network security. In section 3 we provide an overview of the SensorNet with multiple owners, as well as the security model used in this network. In section 4 we discuss the role of security in this network, and then we introduce our scheme. Concluding remarks are provided in section 5.

2. Previous Work

A lot of work has been done on routing in ad hoc networks, as well as providing security in sensor networks. In reference [2] the authors argue that designers need to consider security during the design phase of sensor networks. These authors also note that until security is incorporated in sensor networks at the time of design, these networks will not meet their true potential. In the next few paragraphs we review some of the design decisions that have been made to secure sensor networks.

One approach for securing sensor networks is to use a dynamic key management scheme called Localized Combinatorial Keying (LOCK) [3]. This scheme is limited because it assumes a hierarchical network. It is not clear from this paper how one might manage keys in long-lived sensor networks.

Game theory has also been proposed as an approach for securing wireless sensor networks [4]. With this approach, sensor nodes build a utility function that takes into account the cooperation, reputation and quality of security of several nodes.

Another method for securing sensor networks is proposed by Perrig et al. in reference [5]. This paper

introduces schemes called the Secure Network Encryption Protocol (SNEP) as well as the micro Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol (μTESLA). According to the authors these schemes are necessary since most sensors available at the time of the paper's writing did not have sufficient memory to store a public/private key pair, or perform encryption based on such a key pair. In our opinion μTESLA is hampered by requiring loose time synchronization between the base station and its nodes -- time synchronization might be problematic for cheap sensor nodes. In addition, μTESLA releases the key used in a given interval after a certain period. As a result all nodes that are along the data path can eventually decrypt all the data that was sent in a given epoch, provided they stored the data.

Another proposal for sensor network security is to use multipath dispersion [6]. This paper calls for this approach since sensor nodes cannot be made tamper-proof. The main issue with the multipath discovery process proposed in this paper is that it appears to be complex, since it uses an N-1 multipath discovery process that requires multiple broadcasts.

Location-based keys (LBK) can also be used to provide security in sensor networks that have relatively stationary nodes [7]. With LBK each node's key is based on the node's location as well as its ID. According to this paper LBK are effective in defeating Sybil attacks -- where one node impersonates several nodes, the identity replication attack, and wormhole and sinkhole attacks. In developing its conclusions, this paper assumes that any attacking nodes cannot compromise an unlimited number of nodes. The paper also assumes the existence of a mechanism to report several bogus authentication requests. While LBK appear promising for use in sensor networks, this paper does not tackle the problem of secure routing appropriately.

Some other researchers have also argued for sensor network security to be developed at the software level [8]. According to reference [8], if security is provided at this level, node-to-node authentication can be used to allow network nodes to prove their identity to each other, while node revocation allows WSN to exclude compromised sensor nodes from the network. The authors also argue that all communications and data processing protocols in a sensor network should be resilient in the presence of a few malicious nodes in the network. Data privacy can be provided in sensor networks by using hop-by-hop encryption. According to these authors it is impractical to provide end-to-end encryption in a sensor network since sensor nodes may not have the computational resources to store the necessary keys. The authors argue, as in [6], that multipath routing is one way of getting privacy and resiliency in a sensor network. However, the authors add that the discovery of multiple disjoint paths from a node to a sink remains an open research issue.

Reference [9] lists the requirements for a secure routing protocol. These requirements include import authorization (that is information is only loaded into the routing table if the information is generated by the node owning that information), source authentication and integrity. It is worth noting that these authors do not think the problem of compromised nodes is critical in a non-military network. The paper also considers the prevention of denial-of-service attacks on sensor nodes to be out of scope since these attacks can take place at the physical layer. The authors then list several security flaws of AODV. These flaws include a node forging a RREP or RREQ message, a node modifying a RREQ before forwarding it (so that other nodes think that it has a fresher path to a sink).

In order to secure AODV, reference [9] assumes the existence of a key management subsystem. When a node originates a RREQ or a RREP using SAODV it does the following: Generates a random number seed, sets the Max_Hop_count field equal to the value of the TTL field, sets the Hash field equal to the seed, then computes the Top_Hash value by applying the hash function Max_Hop_Count times to the seed value. Each time an intermediate node receives a RREP or RREQ message it tests whether or not the Top_Hash value is equal to Max_Hop_Count applications of the hash function. As presented, the SAODV scheme also calls for nodes to apply digital signatures to their messages. A node will only respond to a RREQ if the node can verify the digital signature on that RREQ. Finally, the authors observe that their scheme will not guard against tunnelling attacks in a sensor network. It is worth noting that the authors suggest that some variant of identity-based cryptography should be used for generating each node's public key.

As we have seen from our review of the literature, the delayed release of an encryption key is a suboptimal solution for security in sensor networks. In addition sensor nodes are not ideally suited for carrying out public key encryption. As a result we need to try novel methods of providing security in sensor networks, particularly on resource-constrained computers such as motes. One method of providing security that appears to show a lot of promise is multipath dispersion. In section 4 we will look at how this method can be applied to the SensorNet architecture.

3. Overview of SensorNet with Multiple Owners

3.1 Overview

The goal of the SensorNet project is to develop a scalable, flexible sensor network that would incorporate several organizations with different roles. Since the organizations will have different roles, network and application security needs to be set-up such that the organizations will only be able to carry out those tasks that they can. From the grant proposal document [10] we find that the security layer must include methods for authentication, authorization, security policies, privacy and data integrity. In addition the security layer should be able to provide assured access to the sensor network

assets. Many advances have been made thus far on securing the application layer of the SensorNet. In this paper we discuss how the network layer can be secured in order to enhance the overall security of the SensorNet. In particular we will focus on securing routing between motes as well as securing the transfer of data between motes and the data sink. The scheme proposed in this paper has not yet been implemented, but it is designed such that it will operate on resource-limited nodes.

3.2 Security Model Used in SensorNet

In reference [11] Mauro provides an overview of the security model used for Ambient Computing Environments (ACE). ACE provides the backbone for method calls in SensorNet, consequently we will review the security model for ACE in the next few paragraphs.

ACE uses transport layer security (TLS) for authorization, TLS and AES (Advanced Encryption Standard) for data encryption, and Keynote Trust Management for distributing permissions to clients.

When a user needs to use a given service, the user contacts the server to create a session key using TLS. Once the session key has been created all the datagrams are encrypted using the session key. These session keys are generated by a random number generator. The session key is changed after a certain interval to prevent too much data from being encrypted with one key. Datagrams are protected with a packet key under ACE. The general packet format includes a packet key, a packet initialization vector, as well as a SHA-1 hash of the packet's payload. The packet's payload is subsequently encrypted with the AES algorithm using a 128 bit key. It is worth noting that each packet's data is encrypted with a different packet key, while the packet key and the packet initialization vector are protected with the session key. It should be observed that the SHA-1 hash of the packet's payload does not provide any security since it occurs outside of the encrypted payload.

ACE uses other mechanisms for managing users' access to services but these mechanisms are outside the scope of this paper. The reader is referred to reference [11] to get these additional details.

4. Role of Security

The motes in the SensorNet project use the Adhoc On-Demand Distance Vector (AODV) protocol for routing. AODV builds a routing topology of the network by using Route Request (RREQ), Route Reply (RREP) and Route Error (RERR) packets. When a node has a packet to send to a node, it broadcasts a RREQ packet towards the destination. The RREQ message is propagated in the network until it either reaches the intended destination, or it arrives at a node that has a fresh route to the destination. At this time the node that has the RREQ returns a RREP to the node that generated the RREQ. Routes are stored in a node's routing table until they age out. It is worth noting that AODV only stores one route to any given node [12, 13].

It is desirable to secure the routing in the SensorNet by using multipath routing. Unfortunately AODV, in its present form, does not support the caching of multiple routes between a source and a sink. It is worth noting that a multipath extension to AODV was independently proposed in [14, 15]. With the extension proposed in [14], each node can compute either a link-disjoint or node disjoint path between a source and a sink. The source node then selects the best path to each node based on the path that has the best overall signal strengths.

Unlike reference [14] we introduce a multipath extension to AODV whereby each node uses all the paths that it discovers simultaneously. Each source node will try to find k-disjoint paths between itself and the sink. Next, the source node will transmit data to the sink using all k paths simultaneously. As in reference [5] we divide time in this network into epochs. In each epoch the source node will transmit the session key for the epoch along one of the disjoint paths, while transmitting the data for that epoch along other paths. Following a key change in the next epoch, the session key for that epoch will be transmitted along a different path, while the data will be transmitted along a new set of disjoint paths. This change of paths is made to prevent a malicious user from capturing a single node on the key distribution path and attempting to discover the session key. By changing paths several times, a malicious node will have to capture several nodes throughout the sensor network to get a full picture of the data being transmitted. Another benefit of changing paths periodically is that it evens out the energy consumption across the nodes. No nodes will be required to expend most of their energy continually forwarding data on behalf of other nodes. In addition, we would also get some security by not allowing any one node to get a full picture of the state of the network.

4.1 Computing Node Disjoint Paths

We generate the following diagram based on a similar diagram in reference [14]. Suppose node S wanted to compute a node disjoint path to node D. Node S would broadcast a RREQ to nodes A, B and C. This RREQ will be different from the standard RREQ in that it will contain a field ε, to denote how much energy the node has left, as well as a field, π, to denote the power with which the signal was received. Node S would initialize these fields with a value of one. Upon receiving the RREQ, nodes A, B and C nodes would each multiply the ε and π fields with the how much energy they have left, as well as the power with which the message was received then broadcast the RREQ to their neighbours. Without loss of generality, assume that node E receives the RREQ from A before the RREQ from node B. Node E will forward the RREQ from node A. Upon receiving the RREQ from node B, node E will note that it has just forwarded a RREQ from node S destined for node D, so it

will drop the RREQ forwarded by B. The RREQ will be forwarded in this manner until it gets to node D. Node D will also update the ε and π fields, and then it will respond with a RREP that will be forwarded back to node S. Note that the RREP messages will not have their ε and π fields updated. The node disjoint paths are marked with solid lines in figure 1.

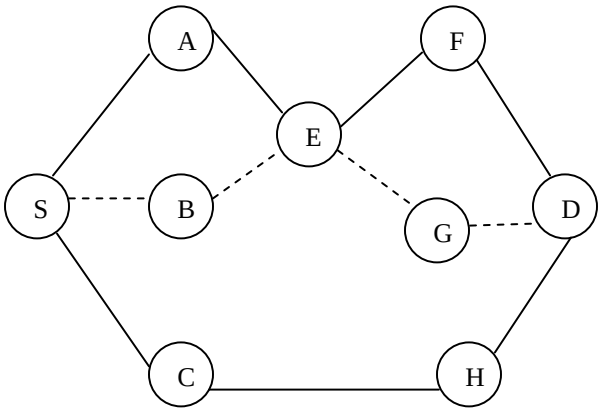

Figure 1: Computing a Node Disjoint Path

At node S, the route metric will be computed as

$$Metric = \frac{Hop\_Count}{\pi * \varepsilon}$$

A lower metric above generally denotes a better path. Once all the paths have been discovered, a node can select the best k paths for use in an epoch. One of the paths should be used for key delivery, while the other paths should be used for data delivery. It also needs to be investigated if the scheme will be improved by doing some form of erasure coding e.g. by XOR'ing the data packets together prior to transmission.

4.2 Using the Scheme to Provide Security

4.2.1 Background

The scheme used in this paper uses a zero-knowledge proof scheme, and potentially identity-based cryptography. As a result, we spend the next few paragraphs discussing those topics.

In reference [16] Fiege et al. develop a zero-knowledge proof scheme. According to the authors this scheme can be implemented on computers with limited processing powers. The scheme is said to be about two orders of magnitude faster than RSA-based schemes. The original paper calls for one to pick a prime number, n that is the product of two primes and of the form $4r + 3$. Once the prime number is computed, the generating center may be closed. Each node then generates k random integers $S_1, S_2,...S_k$. Next each node generates k integers of the form $I_j = \pm 1/(S_j)^2 \ (mod\ n)$ i.e. S_j^2 is the multiplicative inverse of $I_j \ mod\ n$. The I_k integers are published, while the S_k integers are kept secret.

In order to verify information using this scheme, the prover, A, will generate a random number R, and then transform it into a number X. The doubting party, B, will generate a random Boolean vector $(E_1,...,E_k)$ and send that to A. The prover, A, will return Y to B; where Y is some transformation of X. The verifier will then ensure that its value of X matches Y, the transformation of X. The numbers X, Y and the test are as shown below:

Step 1: $$X = \pm R^2 \ (mod\ n)$$

Step 2: $$Y = R * (\prod_{E_j=1} S_j) \ (mod\ n)$$

Step 3 (Test): $$X = \pm Y^2 * (\prod_{E_j=1} I_j) \ (mod\ n)$$

Identity-based cryptography was introduced by Shamir in reference [17]. With identity-based cryptography, a user's public key is composed of his name and network address – or any other combination of identifiers that uniquely identifies him. The private key is generated by a key generation center, and issued to the user. Shamir's original paper is scant on some of the implementation details. However, it is assumed that encryption and decryption operations are carried out in much the same way as with the well-known RSA algorithm.

4.2.1Generation of Random Numbers

Random numbers are critical to the success of this scheme. We argue that random numbers can be easily generated by each node. For example, a node can measure its signal to noise ratio (SNR) over a certain period (epoch), and then use that ratio as an input to a mangling function. The result of this function can be used as a random number for the signing operations. On the other hand, the node can XOR the contents of its registers with the amount of time that it took to fetch those contents. Whichever method is used to generate random numbers, we argue that a node should collect several of these numbers and store them in a file local to the sensor node.

4.2.3 Operation Details

Assume that in our scheme each node sends out a RREQ using the mechanism described above for finding node disjoint paths. In our scheme each RREQ message will also be accompanied by a Boolean vector E. Upon receiving the RREQ a node will respond to the RREQ immediately if it already has a path to the sink, or if it is the sink. If the node does not have a path to the sink, the node will store the Boolean vector E and generate a new Boolean vector F and broadcast a new RREQ. This process will continue until the sink is reached. At this point, the sink will respond to the RREQ with a random

number *X*, a transformation of that random number, *Y*, the Boolean vector, and the route reply (RREP). Note that the random numbers *X* and *Y* are as described above. That response will be verified by the next intermediate node on the path. If that response is verified, the node forwards the RREP back to the previous node on the path. This time the RREP is accompanied by the Boolean vector it received from the previous node, a new random number *X'* and a transformation of that random number, *Y'*. This process of verification and forwarding takes place at each node along the node disjoint path until the source node is reached. At this point, the source node is assured of having a set of node disjoint paths to the sink.

In order to provide for the integrity of routing messages, we propose the use of a hashing algorithm e.g. MD5. Prior to sending a RREP, each node will generate an MD5 hash of the entire message. The node will then generate a random number *Z'* based on a transformation of a random number *R'*. The node sending the RREP will also generate a Boolean vector *E'* and transform *Z'* into the number *Z*. This number *Z* will be used to encrypt the hash of the RREP and then discarded. The RREP will then be sent with the encrypted hash, the Boolean vector, *E'* and the transformation of the initial random number, *Z'*. Each intermediate node will be able to recover *Z* by using the transformation in step 3. Using that result, it will decrypt the RREP hash, and compare that result to a new hash of the RREP. If the two results match, the intermediate node is assured of the authenticity of this reply, and it can then forward the RREP using the parameters from the previous paragraph, as well as a new hash, Boolean vector *F'* and a new key, *A'*.

Step 1: $$Z' = \pm (R')^2 \pmod{n}$$

Step 2 (Key generation): $$Z = Z' * \left(\prod_{E'_j = 1} I_j \right) \pmod{n}$$

Step 3 (Key recovery): $$Z = \pm (Z')^2 * \left(\prod_{E'_j = 1} S_j \right) \pmod{n}$$

The scheme described above allows each node to develop a set of verified paths between itself and a sink. All nodes along these paths are guaranteed to be nodes that were set-up with *n*, the master secret for each sensor network. It can be easily seen from the description above that the trusted paths described above come at the cost of increased processing at each node. It is hoped that a real-life implementation of a zero-knowledge scheme will indeed be as fast as Fiege et *al*. promise, thereby allowing the rapid discovery of a set of routes.

Once the paths described above are formed, each node can then select *k* paths to use for data delivery. Each node can generate a session key per epoch and then select one of the paths for delivery of the session key. During the rest of that epoch no other data should be sent along that path. In the next epoch, a new session key should be generated and sent along a new path, while the path that was used in the last epoch for key delivery should be used for data delivery. Note that unlike the method proposed in reference [5], the session key should be delivered within the epoch and not at some interval after the epoch.

If a node discovers only a single path between itself and the sink node, then that node can use one of two options for data delivery. On the one hand, the node could use a variant of the method described in reference [5]. The node could pick a session key and use that session key to encrypt all data transmitted within the epoch. After a period, δt, after the end of the interval the node can release the session key to the sink. As before δt will be defined to be greater than the round trip time between the sender and the receiver. The other alternative for key delivery in this environment would be to use ID-based cryptography. In this case the source node can pick a session key, and encrypt that session key with the destination node's public key. The encrypted session key will then be delivered to the destination node.

5. Conclusion

This paper reviewed some of the research that has been done in securing sensor networks to date. The paper then reviewed the work that had been done in securing ITTC's SensorNet. Next the paper introduced a new method for exchanging routing information in a secure manner. Future work from this paper includes studying ways of selecting the *k* best paths for routing. We could also study the presence of this protocol in the presence of malicious nodes, and try to establish how much data might be lost if a malicious node successfully implants itself along a path. We could also attempt to simulate the effect the proposed scheme might have on energy consumption at each node. Once all these items have been studied, the routing algorithm will be implemented and tested on the motes.

Acknowledgements

Thanks to Abdul Jabbar Mohammad, Soshant Bali and Weichao Wang for reading and commenting on previous versions of this paper.

References

[1] F. Zhao and L. Guibas, "Wireless Sensor Networks: An Information Processing Approach (The Morgan Kaufmann Series in Networking)," Morgan-Kauffman, 2005.

[2] A. D. Wood and J. A. Stankovic, “Denial of Service in Sensor Networks,” Computer, vol. 35, no. 10, Oct. 2002, pp. 54-62.
[3] M. Eltoweissy, M. Moharrum, and R. Mukkamala, “Dynamic Key Management in Sensor Networks,” IEEE Communications Magazine, April 2006, pp. 122-130.
[4] A. Agah, S. K. Das, and K. Basu, “A Game Theory Based Approach for Security in Wireless Sensor Networks,” Proc. 3rd IEEE International Symposium on Networking and Computing Applications, 2004 (NCA 2004), pp. 259-263.
[5] A. Perrig et al., “SPINS: Security Protocols for Sensor Networks,” Proc 7th International Conference on Mobile Networking and Computing (MOBICOM 2001), ACM 2001, pp. 189-199.
[6] W. Lou and Y. Kwon, “H-SPREAD: A Hybrid Multipath Scheme for Secure and Reliable Data Collection in Wireless Sensor Networks,” IEEE Trans. On Vehicular Technology, vol. 55, no. 4, July 2006.
[7] Y. Zhang et al., “Location-Based Compromise-Tolerant Mechanisms for Wireless Sensor Networks,” IEEE Journal on Selected Areas in Communication, vol. 24, no. 2, Feb. 2006.
[8] H. Chan and A. Perrig, “Security and Privacy in Sensor Networks,” IEEE Computer, vol. 36, no. 10, Oct. 2003.
[9] M. G. Zapata and N. Asokan, “Securing Ad hoc Routing Protocols,” Proc. Of the 3rd ACM Conference on Wireless Security, Sept. 2002.
[10] G. J. Minden et al., “A Unified Architecture for SensorNet with Multiple Owners.”
[11] J. Mauro, “Security Model in the Ambient Computational Environment,” master’s thesis, Dept. of Electrical Eng. & Computer Science, The University of Kansas, 2004.
[12] C. Perkins et al., AODV, http://moment.cs.ucsb.edu/AODV/aodv.html.
[13] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, “Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks,” IEEE Personal Communications, Feb. 2001.
[14] P. Sambasivam, A. Murthy, and E. M. Belding-Royer, “Dynamically Adaptive Multipath Routing based on AODV,” Proc. 3rd Annual Mediterranean Ad Hoc Networking Workshop, June 2004.
[15] E. Biagioni and S. H. Chen, “A Reliability Layer for Ad-Hoc Wireless Sensor Network Routing,” Proc. of the 37th Hawaii International Conference on System Sciences, Jan. 2004.
[16] U. Feige et al., “Zero-Knowledge Proofs of Identity,” Proc. Of the 19th Annual ACM Conference on the Theory of Computing, 1987.
[17] A. Shamir, “Identity-based Cryptosystems and Signature Schemes,” Advances in Cryptology (CRYPTO 1984), LNCS 196, Springer, 1984, pp. 47-53.