This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

A Security protocol for sensor networks

Khadija Stewart Themistoklis Haniotakis Spyros Tragoudas
Department of Electrical and Computer Engineering
Southern Illinois University Carbondale, Illinois 62901
email: jirari@engr.siu.edu haniotak@siu.edu spyros@engr.siu.edu

***Abstract*— We present in this paper a new security protocol especially suited for sensor networks. This protocol uses a novel encryption method for secure message transmission. We present the details of this encryption scheme along with experimental results performed on a network simulator.**

I. Introduction

Sensor networks are large scale, usually slow moving or static wireless networks. The nodes ($motes$) in such networks are designed to sense the environment and collect data. They usually organized into clusters where each cluster is connected to a more powerful base station. These networks have many practical applications which include military use, rescue operations, monitoring and tracking. Security in wireless networks is a big challenge. In fact, wireless nodes are susceptible to multiple kinds of security attacks due to the wireless nature of their links, the limited amount of energy that each node has, their limited processing and storage resources and the absence of any physical protection.

Security attacks on wireless networks are classified as either passive attacks or active attacks. In passive attacks, the enemy nodes try to eavesdrop on the messages exchanged between the nodes in the network without altering them. The purpose of this kind of attack is to secretly obtain the information exchanged between the nodes. In active security attacks, the attackers could replay messages, alter them, or use denial of service where the message is prevented from reaching the destination. Another possible attack is the node take-over attack where one or more nodes are captured and reprogrammed to send falsified readings, claim multiple identities or intercept and/or destroy messages. Active security attacks are particularly harmful towards routing protocols.

In this paper, we focus on the passive attacks. Traditionally, cryptographic methods are used to prevent against eavesdropping. Cryptography can be implemented either in hardware or in software. Hardware implementations are viewed to be more secure and more efficient because they are faster in general and they offer more intrinsic security.

Many cryptographic protocols have been proposed in the literature. Symmetric key and public (asymmetric) key cryptography are the most widely used encryption methods in the area of communications. In symmetric key cryptography, the communicating parties exchange a secrete key that is used for both encryption and decryption of the message (in general, given the encryption key, it is easy to determine the decryption key and vise-versa.) In public key cryptography however, the source (the entity generating the message) uses the public key of the destination to encrypt the message which is decrypted by the destination using the destination's private key. The public key is the encryption key and in this case, knowing the public key does not reveal any information about the private key. Key based cryptographic methods are not well suited for wireless networks because they are very expensive as far as resource requirements, see Section VI.

In this paper, we present an encryption method and its hardware implementation. The proposed encryption scheme partitions the message to be sent into several sub-messages using the encryption method described in Section III. Each sub-message is encrypted using the other sub-messages. The sub-messages are then transmitted to the destination via carefully selected node disjoint paths. The objective is to prevent silent enemy nodes from intercepting all of the sub-messages and decoding them to obtain the original message. All of the sub-messages are needed in order to obtain the original message. Intercepting a few sub-messages would only reveal relationships between the sub-messages and not their actual content. We thus propose a scheme that destroys the correlations in the original message and enforces the secrecy of the data transmitted.

We also give the implementation details of the proposed scheme, examine its hardware requirements and compare them to the requirements of other widely used encryption methods. Furthermore, we prove the strength of the proposed security module in the presence of collaborating intruder or compromised nodes. Intruder nodes are foreign malicious nodes that try to intercept the information exchanged in the network

and compromised nodes are nodes that belong to the network and are captured and reprogrammed by the enemy.

The paper is organized as follows: Section II describes the existing literature on security protocols for wireless networks. Section III gives the details of the encryption and decryption schemes. Section IV describes the scheme for uncorrelating the original message. Section V details the requirements for the routing protocol. Section VI gives the hardware requirements for the proposed security module. Section VII presents the experimental results, and Section VIII concludes.

II. Previous work

In this section, we describe some of the methods used to secure message transmission in wireless networks. In [1], the authors disperse each outgoing message into a number of pieces using the algorithm presented by Rabin in [2] and transmit the pieces along node disjoint paths ($routes$). The algorithm in [2] breaks a message into n pieces so that every $k(k < n)$ pieces suffice to reconstruct the message. This method seems very efficient against denial of service attacks (in denial of service attacks, malicious nodes either refuse to or selectively forward packets) but it seems insecure in the presence of silent enemy nodes. Moreover, Rabin's algorithm relies on computing the inverse of matrices which requires a significant amount of resources that sensor networks can not accommodate.

Also, in [3] the message is broken into separate parts. Half of the message pieces are encrypted and go to one clerk and the encryption key to the other clerk. The same is done with the other half of the message. The clerks (which can be viewed as intermediate nodes) are not able to understand the pieces of the message they receive since the other clerk has the corresponding keys. Any encryption method that uses keys could be used in this case.

In [4], the authors describe a key-management scheme for sensor networks where the keys are predistributed before the deployment of the network. Their protocol relies on probabilistic key sharing among the nodes. Before deployment, each node randomly selects m keys from the pool of keys and stores it in its memory. The total number of keys in the pool is selected such that two random subsets of size m share at least one key with some probability p. At the key-setup phase, a connected graph of secure links is formed by nodes that share at least one common key. [5] is a proposed improvement to the scheme in [4] where $q(q > 1)$ (rather than 1) keys are needed to establish secure communication between a pair of nodes. This new requirement randers the network more resilient to node capture.

[6] and [7] are additional improvements to the basic method proposed in [4] where the resilience of the network is further strengthened.

In [8] the authors propose a security architecture designed for sensor networks. In this scheme, each sensor node shares a secret key with the base station which is then responsible for setting up a trusted key when two nodes need to communicate with each other.

III. The encryption and the decryption method

The proposed security protocol works hand-in-hand with the routing algorithm used in the network. The security scheme operates as follows. When a source node needs to send a message to another node in the network, a set of node disjoint paths is computed by the routing algorithm. Then, the source node splits the message into $2n$ $(n \geq 1)$ parts and each part is routed through a distinct paths (in the case where $n = 1$ refer to [9].) Assuming that message A has $2n * m$ bits $(a_0 ... a_{(2n*m)-1})$ (in case A has $2n * m + k$ bits, $(0 < k < 2 * n)$, we can add $2n - k$ bits as padding), the splitting and encoding of the message is done according to the following steps.

In the first step, the bits of the original message are arranged into a two dimensional array B with $2n$ columns and m rows such that each bit $b_{i,j}$ $(0 \leq i \leq m-1$ and $0 \leq j \leq 2n-1)$ of array B is equal to a_k $(0 \leq k \leq 2nm-1)$ of the original message with $i = k/2n$ and $j = k \ mod \ 2n$, where (mod denotes the modulo operation).

In the second step, another two dimensional array C is created using the encryption procedure $encrypt$ described bellow.

PROCEDURE encrypt (B)
1) for i=0 to m-1 do
   a) $X_i = b_{i,0} \bigoplus b_{i,1} \bigoplus \cdots \bigoplus b_{i,2n-1}$
2) for i=0 to m-1
   a) for j=0 to 2n-1
      i) $c_{i,j} = X_i \bigoplus b_{i,j}$

In the third step, each column of array C is packaged into a data packet and transmitted through one of the pre_determined paths towards the destination node using directional antennas.

Example 1. Let's assume that the message to be transmitted to the destination has 12 bits, $A = 100011101101$. Also, assume that 6 node disjoint paths were computed by the routing algorithm. In this case, $n = 3$ and $m = 2$ since message A will be split into $2n = 6$ sub-messages. The rows of array B are as follows: Row 0 is 100011 and row 1 is 101101. Before encryption, the sub-messages are: 11, 00, 01, 01, 10

and 11. Using procedure $encrypt$, $X_0 = 1$ and $X_1 = 0$. Similarly, array C consists of the following: Row 0 is 011100 and row 1 is 101101. Hence, the encrypted sub-messages are: 01, 10, 11, 11, 00 and 01.

Once all the packets reach the intended destination, the sub-messages are re-assembled using procedure $decrypt$. Note here that we can include a simple redundancy check for the original message to ensure its correctness.

PROCEDURE decrypt(C)
1) For i=0 to m-1 do
   a) $X_i = c_{i,0} \bigoplus c_{i,1} \bigoplus \ldots \bigoplus c_{i,2n-1}$
2) For i=0 to m-1
   a) For j=0 to 2n-1
      i) $b_{i,j} = X_i \bigoplus c_{i,j}$

Example 2. Assume that the sub-messages transmitted are as in Example 1. Using procedure $decrypt$, $X_0 = 1$, $X_1 = 0$, and array B is as follows: Row 0 is 100011 and row 1 is 101101 and the original message A is 100011101101.

Theorem 1: We can reconstruct all of the original parts $b_{i,j}$ where $0 \leq i \leq m-1$, $0 \leq j \leq 2n-1$ and $n > 1$ if and only if all the sub-messages $c_{i,j}$ are given.

The proof is omitted here due to space limitations.

If the encryption/decryption schemes are known, relationships between sub-messages could be deduced if a few sub-messages are intercepted by enemy nodes. We have shown that the amount of information deduced increases as the number of intercepted sub-messages increases. Any such information only gives relationships between the bits of the sub-messages and does not reveal the actual content of the sub-messages as long as not all the sub-messages are intercepted. However, if the bits of the original message are strongly correlated, it could be possible to decipher the original message. An important component in our security protocol is the uncorrelation module that destroys all possible correlations between the bits of the original message. The uncorrelation module is applied to the rows of array B and before the encoding of the sub-messages. As detailed in Section IV, the uncorrelation module destroys correlations between the bits as well as between the rows of array B. In fact, the uncorrelated parts of the original message are pseudorandom.

IV. Scheme for message uncorrelation

We propose the use of Linear Feedback Shift Registers ($LFSRs$) as a mechanism to destroy all possible correlations in the original message and add an extra layer of security, $LFSRs$ are extensively used in the VLSI testing field to produce pseudorandom patterns. "Pseudorandom testing deals with testing a circuit with test patterns that have many characteristics of random patterns" [10]. $LFSRs$ are used to produce test patterns (using either the test-per-scan or the test-per-shift principles) to test for all the possible faults present in digital systems. The test patterns produced must have several properties of random patterns. $LFSRs$ are constructed using memory elements (filp-flops) and simple gates (usually XOR gates.) The feedback connections in the $LFSR$ are dictated by the $LFSR's$ characteristic polynomial. An n bit $LFSR$ with a maximum characteristic polynomial produces $2^n - 1$ different pseudo-random patterns of length n excluding the all-zero pattern. The patterns then repeat with period T=$2^n - 1$.

Thus, before the encryption of the sub-messages, the rows of array B are ran through a maximum-length $LFSR$ for t_i cycles (t_i is randomly chosen for each sub-message, $0 < t_i < T$ and $0 < i < m - 1$). Each sub-message also uses a different characteristic polynomial to destroy the correlations and/or relationships between the sub-messages.

It is however important to append to each transmitted pattern the used characteristic polynomial as well as the t_i variable. This information is encoded using a key (any key based encryption method could be used) for security purposes. Note that the proposed method uses keys. However, the key-related overhead applies to a very small portion of the transmitted information when compared to encoding the overall message using keys.

This last step destroys any correlation that might exist in the original message and renders our security protocol resilient against passive security attacks. Note that given the characteristic polynomial and t_i, the destination node can easily reconstruct the original correlated message.

V. Routing protocol requirements

In the following, we discuss the requirements for the routing algorithm. The main goal of the routing algorithm is to compute node disjoint routes. The constraint that the routes only be node disjoint rather than transmission disjoint is sufficient when using directional antennas. In fact, when using directional antennas, any two node disjoint paths are more likely to be transmission disjoint and with an increased number of paths, the chance of at least two of those paths being transmission disjoint increases.

In this case, each node has to be aware of the approximate position of its neighbor. This could be determined using the Global Positioning System (GPS) or by monitoring the direction of the signals received from each neighbor. A node can overhear a transmission only

if it is within transmission range of the sending node and if is in the angular span of the directional antenna.

The paths could either be computed by the source motes or by the base station and sent to the appropriate sources. For examples of routing protocols that compute transmission disjoint paths see [9], [11], [12] among others.

VI. Hardware requirements

We present in the following the design specifics for the proposed security module. The module is mainly composed of a $2n$ bit LFSR, two levels of XOR gates and three levels of multiplexers. The multiplexers used in our design are $2X1$ multiplexers at the input of every XOR gate and a $4nX2n$ multiplexer to set the initial value of the $LFSR$ memory units. The multiplexers were used as simplified controllers to switch between the encoding and the decoding functionalities of the security module. As shown in Section VII, the power consumption and area occupied by the proposed security module are very negligible.

In wireless sensor networks, the size of a data packet is usually no more than 512 bits. Since each sub-message is packed into an outgoing packet, the packet size will be close to 512 bits. In order to obtain a $2n = 512$ bitwidth module, the transistor count of the security module would be in the vicinity of 10,000 transistors.

VII. Experimental Results

In our experimental results, we give the power consumption and area requirement for the proposed scheme as well as for the RSA and the ECC encryption modules. We also present proof of the robustness of the presented security scheme even in the presence of several collaborating enemy nodes.

We conducted two sets of experiments. In the first set of experiments, we designed the security module using the Very high speed integrated circuit Hardware Description Language ($VHDL$). We then synthesized the module using the BuildGates Extreme Synthesis tool [13] to obtain the power and area characteristics of the circuit. In our experiments, the voltage was set to 1.8 Volts. The power consumption profile and the area for the security module were measured as a function of the circuit's bitwidth. The power consumption and area for the proposed scheme are in the order of a few micro Watts (less than 5) and a few square microns (less than 25) for a circuit capable of processing 512 bits at a time. This provides tremendous power savings compared to the RSA and the ECC chips. According to [14], the power consumption of an RSA chip at 25 MHZ is 500 mW. Also, according to [14], the hardware implementation for the ECC requires a 155 bit block multiplier, which by itself consumes about 300 mW and occupies an area of 24,000 square microns [15].

In the second set of experiments, We used a network simulator to grade the performance of the proposed security scheme. We proceeded with three different phases of experimentations. First, we tested the performance of the proposed scheme in the presence of one intruding/corrupted node. Then, we performed experimental results to determine the number of disjoint paths that offers the best security. Finally, we tested the efficiency of the security mechanism in the presence of several collaborating enemy nodes. We used the OPNET network simulator, version 9.1.A [16].

In the simulation, the wireless nodes were moving at a maximum speed of 20 meters per second, the wireless transmission range was set to 25 meters for each node and the network area dimension was set to 300x300 m^2. We experimented with networks of 60, 70, 80, 90 and 100 nodes and 90 degree directional antennas. The routing algorithm used in this work is a modification of the algorithm used in [9].

In the first phase of experiments, we ran the simulations 10 times for each network size. Each time, a random node was designated as the intruder/corrupted node and the number of paths selected for each trial was the maximum even number of node disjoint paths between the source and the destination. The results of the first set of experiments showed that no individual intruding or corrupted node was able to intercept a complete message. Here we only considered destinations that are at least two hops away from the source node.

For the second phase, we conducted a set of experiments to determine the number of disjoint paths that offers the best security. For each network size, we computed the percentage of intercepted messages when using 2, 4, ..., max_num_paths (where max_num_paths is the maximum even number of node disjoint paths for each particular source/destination pair.) When the number of paths increases, the message is partitioned into more sub-messages and the chance of intercepting all the portions that constitute a whole message decreases.

Table I shows the results for a network of 70 nodes. As expected, when the number of paths increases, the number of intercepted messages decreases for a fixed number of collaborating enemy nodes. The average number of paths for a network of 70 nodes is 6. However, some source/destination pairs have either 2, 4, 6, 8, 10, 12, and up to 14 node disjoint paths. Thus, for example, when limiting the number of paths to 4, the source/destination pairs that only have 2 node disjoint paths only use those two paths.

As a matter of fact, the only limitation of these experimental results is the number of nodes in the network. Sensor networks are usually very dense. These results are pessimistic because the number of node disjoint paths from the source to the destination increases as the number of motes in the network increases. And by carefully selecting the disjoint paths to be further apart, the performance of the proposed security scheme will be drastically improved.

num. collaborating nodes	number of paths			
	2	4	6	Max_num_paths
2	6	3	0	0
4	8	6	2	1
6	10	8	4	2
8	12	8	6	2

TABLE I

THE PERCENTAGE OF MESSAGES INTERCEPTED FOR DIFFERENT NUMBERS OF PATHS FOR A NETWORK OF 70 NODES

The third and final phase of experiments is aimed at testing the efficiency of the security mechanism in the presence of collaborating enemy nodes. For each network size, we randomly designate (2, 4, 6, 8 or 10) nodes to be the intruding nodes. For each number of intruding nodes, we simulate the network 10 times and compute the average percentage of messages that the collaborating intruders were able to intercept.

Table II. shows the results obtained for different network sizes. The first column of the table is the number of nodes in the network and the second column is the even average number of node disjoint routes for each network size. The third column through the seventh column show the percentage of messages intercepted when 2, 4, 6, 8 and 10 intruding nodes collaborate.

num. nodes	num. routes	num. collaborating nodes				
		2	4	6	8	10
60	4	0	2	4	4	4
70	6	0	1	2	2	4
80	8	0	1	3	3	4
90	10	0	1	2	2	3
100	12	0	1	2	2	3

TABLE II

THE PERCENTAGE OF MESSAGES INTERCEPTED IN THE PRESENCE OF DIFFERENT NUMBERS OF COLLABORATING NODES

When enemy nodes intercept a whole message, they need to be aware of the encryption/decryption scheme utilized, the $LFSR's$ characteristic polynomial and the number of cycles for each sub-message in order to be able to decipher the message.

VIII. CONCLUSION

In this work, we present an encryption method designed for use in sensor networks. We show that this method is a perfect fit for the resource constrained sensor motes because of its low power and area requirements. We also demonstrate its efficiency in protecting the secrecy of the messages exchanged in the network through experiments on a network simulator.

REFERENCES

[1] P. Papadimitratos and Z. J. Haas, *Secure Data Transmission in Mobile Ad Hoc Networks*. ACM Workshop on Wireless Security (WiSe 2003), San Diego, CA, pp.41-50, September 19, 2003.

[2] M. O. Rabin, *Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. Journal of ACM, Vol. 36, No. 2, pp.335-348, Apr. 1989.*

[3] Y. Frankel, A Practical Protocol for Large Group Oriented Networks Proc. Of Eurocrypt 1989.

[4] L. Eschenaur and V. D. Gligor, A Key-Management Scheme for Distributed Sensor Networks, in Proceedings of the 9th ACM conference on Computer and Communications Security, Washington, DC, November 18-22, pp. 41-47.

[5] H. Chan, A. Perrig, and D. Song, Random Key Predistribution Schemes for Sensor Networks, in IEEE Symposium on Security and Privacy, Berkeley, California, May 11-14, pp. 197-213.

[6] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, A Pairwise Key Predistribution Scheme for Wireless Sensor Networks, in Proceedings of the 10^{th} ACM Conference on Computer and Communications Security (CCS), Washington, DC, October 27-31, pp. 42-51.

[7] D. Liu and P. Ning, Establishing Pairwise Keys in Distributed Sensor NEtworks, in Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS), Washington, DC, October 27-31, pp.52-61.

[8] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. D. Tygar, Spins: Security Protocol for Sensor Networks, in Proceedings of the 7^{th} Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Rome, Italy, July 2001, pp. 189-199.

[9] Th. Haniotakis, S. Tragoudas, C. Kalapodas, Security Enhancement Through Path Transmission in Ad Hoc Networks, IEEE ICC 2004.

[10] M. Abramovici, M. A. Breuer, A. D. Friedman, Digital Systems Testing and Testable Design. Wiley IEEE Press, 1994.

[11] A. Srinivas, E. Modiano, Minimum Energy Disjoint Paths in Wireless Ad-Hoc Networks. Proceedings of the 9th annual international conference on Mobile computing and networking, San Diego, CA, pp.122-133, 2003.

[12] P. Padimitratos, Z. Haas, Secure Routing for Mobile Ad hoc Networks. Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January 27-31, 2002.

[13] Ehe BuildGates Extreme Synthesis Tool, www.cadence.com

[14] L. Batina, S.B.Ors, B.Preneel, J.Vandewalle, Hardware Architectures for Public Key Cryptography, Integration, The VLSI journal, 34 (2003), pp.1-64.

[15] www.xilinx.com

[16] The Opnet Network Simulator, www.opnet.com