Efficient Security Mechanisms for the Distributed Wireless Sensor Networks*

Prasan Kumar Sahoo, Jonathan Jen-Rong Chen, and Ping-Tai Sun
Department of Information Management
Vanung University
Chung-Li, Taiwan, 320
E-mail: {pksahoo, jonathan, kuei}@msa.vnu.edu.tw

Abstract

In this paper, a secure data communication environment for the three-tiered Wireless Sensor Networks (WSNs) that continues to operate correctly in a hostile medium is proposed. Considering the energy and hardware constraints of the sensor nodes, the low complex data confidential and authentication algorithms are proposed. Performance analysis of our protocol shows that it satisfies the energy and hardware limitations of the WSNs and maintains the secure fabrics of the network.

1. Introduction

The applications of wireless sensor networks [1] widely range from the indoor applications like smart home, health monitoring in a hospital to outdoor applications like highway traffic monitoring, combat field surveillance, security and disaster management. The most important outdoor applications like military surveillance, enemy ship movement and terrorist threats tracking need to check the privacy and security issues. Largely deployed sensor nodes may cover a huge area further exposing them to attackers who may capture and reprogram the individual nodes. The adversary may use its own formula of attacking and induce the network to accept them as legitimate nodes. Falsification of original data, extraction of private sensed data, hacking of collected network readings and denial of service are also certain possible threats to the security and the privacy of the sensor networks. Though hardware and software improvements may address many of such security issues, but development of new supporting technologies and security principles are challenging research issues in WSNs.

The symmetric key sharing among the nodes of the network is an important design issue for the security protocols. Though this key sharing approach has the lowest storage costs and very energy-efficient, but there are obvious security disadvantages such as the compromise of a single node will reveal the global key. In the other hand, the sharing of keys pairwise between two nodes is more ideal since the compromise of a node does not reveal any keys. However, in this approach each node requires a unique key and keying relationship needs to be established after the network is deployed. Another design security issue in WSNs is to maximize the lifetime of sensor nodes. So, computation and operations of nodes during possible security verifications should be energy efficient and satisfy the hardware constraints. In this paper, we propose a security mechanism for the WSNs and the main contributions are:

- To analyze security challenges and the respective implementation feasibilities in WSNs.
- To design a secure architecture for the three-tiered WSNs.
- To propose a protocol that supports three types of keys for the whole network.
- To design a low complex data confidential algorithm.
- To develop a low computational overhead based authentication algorithm.

The rest of the paper is organized as follows. Backgrounds of the sensor networks security is discussed in Section 2 and related works are given in Section 3. System model of our protocol is presented in Section 4 and our security protocols are presented in Section 5 of the paper. Performance analysis of our protocols is made in Section 6 and Conclusions are drawn in Section 7 of the paper.

*This work is supported by the National Science Council of Republic of China under grant NSC 93-2213-E-238-009.

Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)
0-7695-2316-1/05 $20.00 © 2005 IEEE

2. Backgrounds

2.1 Security Challenges

In the 21st century, advance in computing and communications has made a dramatic change in sensor research in the areas of computing, communication and sensing. Since sensor network communication is based on broadcasting, there are every possibilities that attacker can eavesdrop the message and reply it. In a sinkhole attack, adversary tries to attract nearly all the traffic from a particular area and creates a sinkhole in the network. It causes the routing algorithm to attract other nodes to send their data through it, manipulate the data and then sent to the Base Station (BS). In the Sybil attack [2], a single node presents multiple identities in the network to put other nodes in trouble. In the Wormhole attack [3], the adversary tunnels messages received in one part of the network over a low latency link and replays them in a different part. It creates sinkhole in the network and the shortest route among the nodes to the base station to intercept the message. This attack may be used in combination with selective forwarding or eavesdropping. Another potential attack is the Hello Flood Attack and it is similar to the broadcast Wormholes attack. In this attack, it uses a single hop broadcast to send a message to a number of receivers.

2.2 Implementation Feasibility

In WSNS, there are several constraints to implement the standard security algorithms as they are designed for the powerful workstations. Besides, the sensor nodes have very limited computational and communication resources for many arithmetic and logical operations. Due to the hardware constraints of sensor nodes, the public key certificates in asymmetric cryptographic algorithms like RSA [4] and Diffie-Hellman [5] are not suitable for WSNs as the working memory of a sensor node is insufficient even to hold the variables. The symmetric algorithms, like AES and integrity/authentication algorithms, like HMACs [6] incur high computational energy costs. So, the core asymmetric, symmetric and authentication algorithms are not suitable for WSNs as the computational cost is an overhead to the power consumption.

Based on different hardware constraints and the applications of WSNs, we have classified the sensor nodes into three categories such as the generic, special-purpose and the high-bandwidth sensors. The hardware specifications of these nodes are given in Table 1, 2 and 3 respectively.

3. Related Work

Several symmetric algorithms [7, 8] have been proposed for broadcast authentications. But, such algorithms are not suitable for WSNs due to high communication overhead per packet. The security Protocols for Sensor Networks (SPINS) [9] such as SNEP and μ-TESLA has been proposed for the resource constrained WSNs. Several key exchange, distribution and management protocols [10, 11] have been proposed for the pre or post deployed sensor nodes. In LEAP [15], a key management protocol for the sensor networks that support the in-network processing is proposed. This protocol supports the establishment of four types of keys for each sensor nodes and are used for establishing and updating the keys and simultaneously minimizes the involvement of the base station. In SEKEN [11], a scalable, power efficient secure protocol is proposed. This protocol allows each sensor node to share two types of keys e.g. a master key shared with the base station and an explicit key between individual neighboring nodes to exchange the secure information.

Processor	8-bit, 4 MHz
Memory	8 KB flash, 512 RAM, 512 Bytes EEPROM
Radio	916 MHz
Data Rate	10 Kbps

Table 1. Prototype of generic-sensor nodes(Mica Mote)

Processor	4-8 Mhz Custom 8-bit
Memory	3K-4Kb RAM, 0.1 Mb flash memory
Radio	50-100Kbps
Data Rate	20 Kbps

Table 2. Prototype of special-purpose sensor nodes(Spec 2003)

4. Systems Model

We describe here a three-tiered system model for the wireless sensor network comprising the Sensor nodes(SN), Gateway nodes(GN) and Base stations(BS) as shown in Figure 1. We divide the whole network into certain clusters and each cluster comprises one GN that controls several SNs. The GNs of different cluster communicate with each other

Processor	Intel StrongARM 1100@133 MHz, 150 MIPS
Memory	1MB SRAM, 4 MB Flash memory
Radio	3 wire RS-232
Data Rate	100 Kbps

Table 3. Prototype of high-bandwidth sensing nodes(RSC Wins-Hidra Nodes)

to exchange the collected data. The GNs forward the collected data to the nearby BS and finally to the user or the controlling authority, which is located somewhere, far away from the monitoring region that access the sensed data and monitors the network via the BSs. The three different tiers of the WSNs may be planned as given below.

Tier-1: These are the set of generic sensor nodes(SN) like Mica Motes[16] and are deployed hundreds of thousands in a specific monitoring area. The whole monitoring area is divided into certain clusters which can be formed based on many criteria such as communication range, geographical location and based on the number and type of sensors for different applications [12, 13, 14]. Their functions are simple, specific and are usually operated independently. They sense the medium, collect the raw data and forward it to the next hop neighbor nodes and ultimately to the second tier. The hardware specifications of such nodes are shown in Table 1.

Tier-2: These are some special-purpose sensor nodes like Spec 2003[16], limited number of which are deployed in the monitoring region. In each cluster, there exists only one cluster head and is termed as the Gateway node(GN), which can collect raw data from the SNs of its cluster. These nodes are more powerful in computation and energy than the SNs and their respective prototypes are presented in Table 2. Each GN of the network has unique ID and its assignment is based on the cluster number. GNs can track events or targets using the sensors of its own cluster and prepare the final report using data fusion and aggregation techniques and forwards the fused data to the third tier.

Tier-3: The high-bandwidth sensing and communication nodes like RSC Wins-Hidra Nodes[16] form the third tier of the network and are known as the BS of the WSNs. The operating characteristics of such nodes are given in Table 3. These nodes have relatively powerful processing, memory and transmission capacity and are having long battery life. These BSs and the user or the controlling center are connected via wireless such as internet and satellite.

5. The Security Protocols

In this section we propose three types of keys that are used during necessary security verifications and are described below. Our algorithms for the data confidentiality and authentication are applicable to all types of nodes in the network in a distributed fashion, irrespective of its presence in any particular tier.

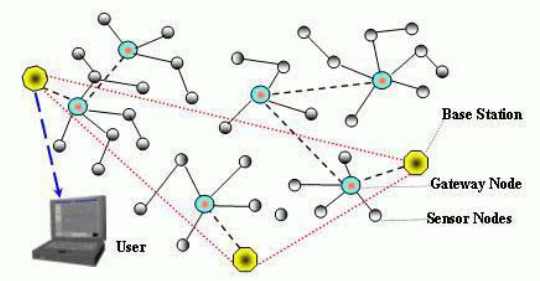

Figure 1. The three-tiered architecture of WSNs

5.1 Overview

In our protocol, we assume that packet transmission among the SNs in each clusters e.g. the nodes in the first tier is only broadcasting and the routing of the packets among the GNs and GNs to BS e.g. within the second and third tiers are only unicasting. We propose three different types of keys for the whole network, which is summarized below.

Sensor Key: Each sensor nodes in the networks has a unique secret key that it shares with other SNs of the same cluster and is denoted as E_{SN}. This key is same for all the SNs of the whole network. We assume that the SNs of each clusters are fixed with respect to their cluster head (GN).

Gateway Key: Each cluster head, that is otherwise known as GN has a unique secret key which is denoted as E_{GN}. It is shared by the GNs of other clusters. It is to be noted that the sensor node key, E_{SN} and the gateway node key, E_{GN} are different from each other.

Base Station Key: Each BS in the network has a unique secret key that is shared among the BSs and is denoted as E_{BS}. This key is distinct from the E_{SN} and E_{GN}.

In this protocol, we assume that each SN of the whole network stores its own secret key E_{SN} for necessary security verifications among the SNs and the GNs secret key E_{GN} for necessary security verifications of SNs with the GNs. We assume that the data flow in WSNs is from SNs to the GNs as sensor nodes are meant to probing the environment to detect a target or event. So the GNs use the E_{SN} to maintain the data confidentiality between its own and the

sensor node from which it receives any packet. Under special circumstances, if a GN issues mission, sends queries and interests, the SNs those are immediate neighbors to that GN and receive the packets, use the shared key E_{GN} for perusing the data confidentiality. We assume that the secret keys of SNs, GNs and BSs are assigned at the manufacturing phase as the key assignment in the network formation phase in a hostile medium is not secure. Besides, we assume that the nodes in the second tier e.g. the GNs store three different types of keys, E_{SN}, E_{GN}, and E_{BS} for necessary security verifications with the SNs, other GNs and with the BSs respectively. It is to be noted that since each SNs, GNs or the BSs has the same shared key, post deployment of nodes won't have any effect in establishing the shared keys relationship. For example, sensor nodes A, B and C can use their shared key whether they are in the same or in different clusters after the deployment of nodes as all the shared keys for every sensors are same.

5.2 Data Confidentiality

Before we describe the data confidentiality algorithms, we present here some of the useful terms that we have used in the next subsequent steps.

Let A and B are two different communicating SNs present in the first tier of the network and they transmit *x*-bits of message *M*.

n is a pre-assigned integer such that $0 < n^2 < x$ and represents the number of rows and columns of any matrix. It is known to both the sender and the receiver in advance e.g during the network construction phase and the value is fixed for all the messages.

Divide the whole message *M* into *k*-numbers of sub-messages M_1, M_2,.....,M_k of n^2-bits each such that $M_1 \mid M_2 \mid ..... \mid M_k$ denotes the concatenation of *k*-numbers of message into *M* where *k* = $\lceil \frac{x}{n^2} \rceil$.

E_{SN} is the secret key of n^2-bits which is shared between sensor nodes A and B and $\odot$ is a binary operation like XOR or XOR-NAND. Both E_{SN} and $\odot$ are known to both the nodes in priori. However, $\odot$ acts like a session key and type of operation can be changed time to time by the cluster head GN, to make the security more stronger.

$\{M\}_{E_{SN}}$ is the cipher message of message *M* using the secret key E_{SN}.

Suppose a message *M* of n^2-bits is sent by any node, after encrypting it by the n^2-bits secret key E_{AB}. Then the encrypted message is:

$\{M\}_{E_{SN}} = M_{SN} \star E_{AB}$ *(modp)*, where *p* is a prime number of order 512 bits in the Galois field.

Though E_{SN} is the secret key and only known to the sender and the receiver, there is possibility that it can be broken. Because, if hacker can intercept the message of any packets sent to the destination, it is possible that adversary can get knowledge about the secret key as adversary easily hacks the cipher message and use the following rule to get the secret key.

$M^{-1} \star \{M\}_{E_{SN}} = M_{SN} \star E_{AB}$(modp)

Once, the adversary knows the secret key from any of sent packet, it'll be easier for it to break the confidentiality for next subsequent packets. In order to overcome such problem and considering the technical constraints of sensor nodes, we modify the above idea to make data confidentiality more stronger. We propose to generate a new encrypted key by using the physical situations like time of the sent message or temperature at the time of sent message etc. This information can be sent either in the control packet or in the data packet containing the message.

In our algorithm, we have considered a time stamp matrix T_i of n^2-bits, for $\forall$ *i=1,2,3,...,k.* For each message, sent at different instant of time, different time stamp matrix T_i and the shared key E_{SN} are used to generate a new encrypted key ${E_i}^{SN}$which is only known to the sender.

Thus the subsequent new secret keys can be generated as follows:

$${E_1}^{SN} = E_{SN} \odot T_1$$
$${E_2}^{SN} = E_{SN} \odot T_2$$
$${E_3}^{SN} = E_{SN} \odot T_3$$
.....
$${E_k}^{SN} = E_{SN} \odot T_k$$

where, T_1, T_2, T_3, ..., T_k are the time stamp matrices which are based on the local time at which a message is sent and $\odot$ is an operation that is only known to both sender and the receiver. This $\odot$ acts like a session key between the SNs and the GNs and GN for a cluster updates the type of operation time to time.

Now break the original message *M* into *k*-number of messages $M_1, M_2, ..., M_k$, each of n^2-bits. Since, *k* = $\lceil \frac{x}{n^2} \rceil$, it is obvious that for $\frac{x}{n^2}$ is not a whole number, *M* can be broken into *(k-1)* number of messages of n^2-bits each, and another one message (M_k) of $[x\text{-}n^2 \star (k\text{-}1)]$-bits, which is less than n^2-bits. In this case, the last message M_k will have $[x\text{-}n^2 \star (k\text{-}1)]$-bits of message and rest bits are garbages such as ♡ or anything else. However, if $\frac{x}{n^2}$ is a whole number, *M* is broken into *k*-numbers of messages, each having n^2-bits. The new cipher message C_i is generated by taking the messages M_i and the new encrypted matrix ${E_i}^{SN}$. Thus the

transmitted cipher message at different instants T_1, T_2, T_3, ..., T_k are:

$C_1 = M_1 \star E_1{}^{SN} (\text{mod p})$

$C_2 = M_2 \star E_2{}^{SN} (\text{mod p})$

.....

$C_k = M_k \star E_i{}^{SN} (\text{mod p})$

Finally, the sender transmits the original message *M* in form of the cipher messages C_i. The data packet transmitted by the sender contains n^2-bits of the cipher message and n^2-bits of time-stamp matrix and the whole message *M* is transmitted for *k*-times. On receiving the data packets, the gateway nodes decrypt each cipher messages using the shared secret key E_{SN} and the time-stamp matrix T_i. Since the shared key E_{SN} and binary operation $\odot$ are only known to the sender and the receiver, the data confidentiality can't be lost even though hacker receive the message. Thus, similar principle can be applied for establishing the necessary data confidentiality between the SNs and the GNs. However, when the GN receive the data packet from any SN, it uses its secret key E_{SN} instead of E_{GN}, to decrypt the message. But, if any message is sent from one GN to other, it uses E_{GN} to encrypt it before sending to other GN of the network.

5.3 Authentication

We propose here a low complexity public key encryption which is applicable to all the three tiers of the network. We assume that each cluster head GN will assign a unique SN ID to each of the sensor nodes present in a cluster and maintains the ID information of those sensors. Similarly, each GN will have a unique ID such as the ID of the cluster that it belongs to and each BS will have a unique ID too. It is to be noted that, in our protocol, we consider the GNs are having more energy and hardware capabilities than the SNs. So maintenance of ID of the SNs of a particular cluster won't be a burden for the GNs. The ID of either the SN or GN or BS is taken as the public key for the authentication verification, details of which are as follows.

Let *y*: ID of the SNs/GNs/BS, is the public key.

m: The cipher message, encrypted as per the data confidentiality technique, described in the previous section.

a, b: Unknown variables

x: Sender's private key

Now the sent message from A to B is:
A→B: A(*y,a,b,m*) and the cryptographic function is:

$x^2 \equiv y \ (mod\ n)$ such that

$a\text{-}b \equiv (m+1) \star \frac{x}{\alpha} (mod\ n)$

$a+b \equiv (m^2\text{-}m+1) \star x\alpha (mod\ n)$

where α is a random number s.t. $\alpha \in Z^*{}_n$ and *n* is composite number of 1024 bits. On receiving the the packet containing *y, a, b and m*, receiver B can calculate $a^2\text{-}\ b^2 \equiv (m^3+1) \star y (\text{mod n})$.

Ultimately, node B uses the public key cryptographic mechanism to calculate the value of *n*. If it matches with its preserved value of *n* with A's value of *n*, then it authenticate A as a legitimate node. It is to be noted that *y* is the ID of the sender and for each sender there will be a unique *n* that should match with the receiver's *n*.

6. Performance Analysis

In this section we analyze the computational and storage cost of our protocol due to the key updating, establishment, encryption and decryption operations during the confidentiality verification. It is to be repeated here that in our protocol, we don't need any key updating mechanism as we assign a single key to all the sensors, another single key to all the GNs, and also a single shared key to all the BSs of the network. So in our protocol, there is no computational cost required in establishing the keying relationship among either the SNs or GNs or BSs. Also, our protocols don't impose any computational burden for key updating or in establishing the keying relationship. However, the computational cost in encrypting or decrypting the message can be calculated as follows.

In case of SNs: Suppose, in a cluster a node has n different neighbors and x_i, *i=1,2,3...,n* be the number of neighbors of those n nodes. So total number of required encryptions is: $E_T = \sum x_i$, for *i= 1,2,3...,n*. Similarly total number of decryption is also $D_T = \sum x_i$, for i= 1,2,3...,n.

In a cluster, average number of symmetric operations are $= \frac{2\sum x_i}{(n+\sum x_i+1)}$

In case of GNs: In our protocol GNs communicate with each by unicasting the message. Suppose, the whole network has m numbers of GNs. In the worst case, a GN will have at most *(m-1)* neighbors. The average number of encryptions and decryptions in case of the GNs is = $\frac{2(m-1)}{m}$

In case of BSs: Suppose, the whole network contains *p* number of BSs. As the communication among the BSs is also unicasting, average number of encryptions and decryptions is = $\frac{2(p-1)}{p}$.

In our protocol, $p < m < n.$. So the total average number of encryption and decryptions operations = $\frac{2\sum x_i}{(n+\sum x_i+1)} + \frac{2(m-1)}{m} + \frac{2(p-1)}{p}$. Besides, in our protocol, a node stores only two types of keys e.g . E_{SN} and E_{GN} and keys are same for all the nodes the cluster. So there is no requirement to store the chain of keys for its neighbors. If l_1 is the key length of E_{SN} and l_2 is the key length of E_{GN}, then the total key length is required to store in each SN is $l = l_1 + l_2$. Though, memory space is the scarce resource for the sensor nodes, for a reasonable key length of E_{SN} and E_{GN}, storage is not an issue in our protocol. It is observed that the storage requirement, encryption and decryption computational costs of our protocol is better than the LEAP [15].

7. Conclusion

We propose here the data confidential and authentication algorithms for a three-tiered WSNs. We implement three types of keys to minimize the storage capacity and computational cost. In future, we'll implement the algorithms and perform some experiments to verify the energy consumption of our protocols. However, from the theoretical analysis, our algorithm is suitable for the WSNs within its present constraints and we demand that it can be applicable in the hostile environments like battle field to locate the movements of the enemy or to detect the terrorist threats.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks", *in IEEE Communications Magazine*, vol. 40:pp. 102-114, Aug 2002.

[2] J. R. Douceur, "The Sybil Attack," *in 1st International Workshop on Peer-to-peer systems (IPTPS 02)*, March 2002.

[3] Y. C. Hu, A. Perrig, and D. B Johnson, "Wormhole detection in wireless ad hoc networks," *Department of Computer Science, Rice University, Tech.Rep*,TR 01-384, June 2002.

[4] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *in Communications of the ACM*, 21 (2): pp. 120-126, 1978.

[5] W. Diffie and M.E.Hellman, "New directions in cryptography", *in IEEE Trans. Inform. Theory*, IT-22:644-654, November 1976.

[6] R. Gennaro and P. Rohatgi, "How to sign digital streams in Burt Kaliski, editor", *Advances in cryptology-Crypto* '97, pp. 180-197.

[7] A. Perrig, R. Canetti, J.D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels", *in IEEE Symposium on Security and Privacy*, May 2000.

[8] P. Rohatgi, "A compact and fast hybrid signature scheme for multicast packet authentication", *in 6th ACM Conference on Computer and Communications Security*, November 1999.

[9] A.Perrig, R. Szewczyk, V.Wen, D.Cullar, and J.D. Tygar, "Spins: Security protocols for sensor networks," *in Proceedings of the 7th Annual ACM/IEEE Internation Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, July 2001, pp. 189-199.

[10] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge", *in Proc. of IEEE INFOCOM'04*, Hongkong, March 2004

[11] K. Jamshaid, and L. Schwiebert, "SEKEN (Secure and Efficient Key Exchange for Sensor Networks)", *in the 23rd IEEE International Performance Computing, and Communications Conference (IPCCC)*, April 2004.

[12] A Buczak and V. Jamalabad, "Self-organization of a Heterogeneous Sensor Networks by Genetic Algorithms," *in Intelligent Engineering Systems Through Artificial Neural Networks*, C.H. Dagli, et.al.(eds.), Vol.8, ASME Press,1998.

[13] C. Richard Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks", *in IEEE Journal on selected areas of communications*, 15 (7), September 1997.

[14] C.-M Liu and C.-H. Lee, "Power-efficient Communication Algorithms for Wireless Mobile Sensor Networks", *in ACM PE-WASUN'04*, October, Venezia, Italy, 2004.

[15] S. Zhu, S. Setia and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks", *in the 10th ACM Conference on Computer and Communications Security (CCS '03)*, Washington D.C., October, 2003.

[16] The hardware specifications: *http://www.cse.unsw.edu.au/ sensar/hardware/hardware survey.html*