國立交通大學

資訊科學與工程研究所

碩士論文

在無線微感測網路中安全的
能源平衡路由通訊機制

An Energy-Balancing Secure Routing Scheme in Wireless Sensor Networks

研究生：劉士豪

指導教授：謝續平 教授

中華民國 九十五 年六月

無線微感測網路中安全的能源平衡路由通訊機制

An Energy-Balancing Secure Routing Scheme in Wireless Sensor Networks

研究生：劉士豪　　Student：Shih-Hao Liu

指導教授：謝續平 博士　　Advisor：Dr. Shiuh-Pyng Shieh

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

無線微感測網路中安全的能源平衡

路由通訊機制

研究生：劉士豪　　　　　　　　　　　指導教授：謝續平 博士

國立交通大學資訊工程學系碩士班

摘要

隨著無線微感測網路的廣泛使用，安全路由通訊協定的重要性也隨之增長。在現有的安全路由通訊協定當中，大部分的路徑選擇方法會選擇對單一路徑能源消耗最少的路徑，但是對整個無線微感測網路來說，這類選擇路徑的方法往往會導致網路中某些感測節點很快的消耗完其極有限的能源。因此，當網路中發生因為路由路徑分布所導致的能源消耗不平均現象時，將會在短時間內降低網路的連結性，甚至使網路呈現非連結的狀態。因此，在本論文中，我們針對無線微感測網路的能源消耗情形，提出了一個新的安全路由通訊協定。此安全路由通訊協定不但提供了資料完整性、資料機密性和鄰居認證機制，更重要的是能夠透過基地台所得到的資訊來調整路由路徑，以平衡每個網路節點的能源消耗，以延長整個網路處於連結狀態的總時間。最後，在本論文中，我們亦依所提出之路由通訊機制，進行無線微感測網路能源消耗及網路連結狀態進行模擬分析。模擬的結果顯示我們所提出的安全路由通訊協定在整個網路在連結狀態的總時間上，比現有的省電安全路由通訊協定多出三至四倍，也比傳統無安全性的低能源路由機制高出十至五十個百分比。

An Energy-Balancing Secure Routing Scheme in Wireless Sensor Networks

Student: Shih-Hao Liu **Advisor: Dr. Shiuh-Pyng Shieh**

Department of Computer Science and Information Engineering
National Chiao-Tung University

Abstract

As the number of wireless sensor network applications increase, there is a greater emphasis on designing secure routing protocol. Most recent secure routing protocols tend to find the minimum energy cost path to lower energy usage for all routing paths, thus, they may lead to some of the nodes deplete their energy quickly. When energy consumption cause by routing is unbalanced over the network, the network would loss connectivity even fall apart in a short time. In this paper, we introduce a new secure routing scheme that not only provides data confidentiality, neighbor authentication, and evidence of data freshness but also balances energy consumption of each node using a base station to adjust the routing paths in order to prolong the total time of overall network connectivity. Simulation results show that the total time when network is connected increases 3 to 4 times over our related work of secure routing schemes with minimum energy cost path, and up to 10 to 50% over the related work of the conventional energy aware routing scheme.

誌謝

首先感謝交通大學網路安全實驗室在我兩年研究生生活中給了我許多寶貴的經驗以及知識，感謝在這兩年中，指導老師謝續平教授給我的諄諄教導，教導我如何做研究，同時也感謝實驗室的學長、同學以及學弟們的互相討論、切磋，讓我的研究可以更完整。另外父母的全力支持讓我能無後顧之憂的專心做研究，非常感謝他們。

Table of contents

List of Figures

1. Introduction

Wireless sensor networks have important potential applications such as virtual fences [2], environmental monitoring [1][4], military [5], health care [3][8], surveillance [7], and security system [6]. The main propose of wireless sensor network is to gather information about the environment or object they are sensing and send these information to the back-end base station. In wireless sensor network, each sensor node has the capability of information sensing, data processing, wireless communicating, and operates on its limited batteries.

Compared with general ad hoc networks, wireless sensor networks have some unique characteristics due to their hardware limitations. The features of wireless sensor networks are:

- Critical of energy consumptions: Most wireless senor nodes are powered by batteries, which possess restricted energy, are difficult to be recharged. Energy consumption is becomes one of the most important challenges in wireless sensor networks.
- Low communication bandwidth: Radio-frequency transmission is the only channel to transmit data, but it is the main power consumption on a node. Due to the limitation of energy capacity of a node, the bandwidth of its transceiver is about 20 to 150 kbps.
- Limited computation power and memory space: Wireless sensor devices tend to be made for low-cost purpose, so its computation power and memory space are critically designed. There is only several kilo bytes memory equipped on each node, and the computation power is about 4 to 40MHz.
- Highly damageable environment: Due to the low-cost design of wireless

sensor devices, tamper resistance is beyond the concept. Each node is highly damageable under many external forces from the deploying environment.

Pivotal to the success of wireless sensor network is the appearance of tiny, lightweight, network devices, called MICAz. MICAz is a low-power, tiny wireless measurement system which is designed specifically for deeply embedded sensor networks. Its wireless communication transceiver follows IEEE 802.15.4/ZigBee spec with 250 kbps data rate. The program memory is 128K bytes while data memory is 4K bytes. Power consumption of microprocessor is 8mA in active mode and less than 15 μA in sleep mode while that of radio-frequency transceiver is 19.7mA in receive mode, 11mA in transmit mode, 20 μA in idle mode and 1 μA in sleep mode.

Trying to network a large number of such low-power wireless devices is a great challenge and has been the focus of many researches. We focus on secure routing problem, especially on how a secure routing protocol affects the total time of overall network connectivity.

Most prior works on secure routing try to find the minimum energy cost paths to reduce the energy consumption of each routing path. SEAD (Secure Efficient Distance vector routing protocol) [10], which is based on DSDV (Destination-Sequenced Distance-Vector routing) [11], is a hop-by-hop distance vector routing protocol requiring each node to periodically broadcast routing updates. The key advantage over traditional distance vector routing protocol is that it guarantees loop-free minimum hop count paths. Ariadne (A Secure On-Demand Routing Protocol for Ad Hoc Networks) [14] uses source routing rather than hop-by-hop routing, with each packet to be carrying in its header a complete, ordered list of nodes through with packet must pass. Intermediate nodes do not need to maintain up-to-date routing information, but the cost of source node establishing the routing path is high due to flooding whole network by route request message. It tends to find the paths with minimum traveling time. ARAN [13] and SAODV [15] are securing routing protocols based on AODV (Ad-hoc On-Demand Distance Vector Routing) [12] which is essentially a combination of both DSR (Dynamic Source Routing) [17] and DSDV. It uses the concept of route discovery and route maintenance of DSR and hop-by-hop routing vector, and periodic beacons form DSDV. These AODV based secure routing protocols have the same feature that each routing path is minimum hop count and is energy efficient from the point of view on each routing path. However, because of minimum energy cost selecting feature, some of the nodes with best score would carry a lot routing paths. As pointed out in "Energy conserving routing in wireless ad-hoc networks" [9], this feature can lead to some nodes in the network being drained out of energy very quickly.

In this paper, we are going to propose a new secure routing scheme that balances the energy consumption of each node using a base station to adjust the routing paths, and it provides data confidentiality, neighbor authentication, and evidence of data freshness for security. We will show that the proposed scheme is able to increase the total time of overall network connectivity efficiently.

In wireless sensor networks, base station is the single destination that every source node is going to send its sensed information to. So base station can get information about whole network collected from each route. According to the collected information, base station adjusts some of the nodes in the network to change the load distribution in order to balance energy consumption of the whole network.

In the proposed scheme, each node can balance the energy consumption of overall network caused by routing without knowing global information and only one network traversing is needed instead of maintaining the whole network status all the time. Data confidentiality, neighbor authentication, and evidence of data freshness are supported for data forwarding over the wireless environment in secure manner.

The rest of this paper is organized as follows. Some prior works about secure routing will be described briefly in section 2. We introduce energy-balancing secure routing scheme in detail in section 3. Evaluations for comparing energy-balancing secure routing over other secure routing protocols will be given in section 4. Security analysis presents in section 5 before concluding this paper in section 6.

2. Related Work

We will introduce prior works in secure routing in section 2.1 and SEAD will be introduced in section 2.2. In section 2.3, we will give a rough description of a secure on-demand routing protocol for ad hoc networks (Ariadne).

2.1. Secure Wireless Ad Hoc Routing

Secure ad hoc network routing protocols are difficult to design, due to the generally highly dynamic nature and due to the need to operate efficiently with limited resources. Current designs of secure routing schemes in wireless ad hoc networks are based on existing insecure ad hoc network routing protocols. According to different routing techniques, secure routing schemes can be classified into several categories.

DSDV-based Yih-Chun Hu and Adrian Perrig proposed SEAD [10] which is robust against multiple uncoordinated attackers creating incorrect routing state in any other node, in spite of active attackers or compromised nodes in the network. SEAD is designed in part on the Destination-Sequenced Distance-Vector ad hoc network routing protocol (DSDV) [11]. We will go into detail about SEAD later.

DSR-based Dynamic Source Routing (DSR) [17] is proposed by David B. Johnson and David A. Maltz. It is an on-demand routing protocol, which the source nodes need to maintain the whole routing paths to each destination. Yih-Chun Hu and Adrian Perrig proposed a secure on-demand routing protocol for ad hoc networks (Ariadne) [14], which is a DSR based secure routing. It withstands node compromise and relies only on highly efficient symmetric cryptography. We also will introduce Ariadne briefly later.

AODV-based Ad-hoc On-Demand Distance Vector Routing (AODV) [12] is proposed by Charles E. Perkins and Elizabeth M. Royer. AODV provides minimum hop-loop free routes by periodically broadcasting advertisements for spreading distance vector routing information to each node. Kimaya Sanzgiri and her colleagues developed authenticated routing for ad hoc networks (ARAN) [13], which is based on AODV. In ARAN, each node has a certificate signed by a trusted authority, which associates its IP address with a public key. ARAN is an on-demand protocol, breaken up into discovery and maintenance. Figure 2-1 shows an example of route discovery in ARAN while figure 2-2 shows that of route maintenance.

In figure 2-1, node S is discovering a route to node D. Each node rebroadcasts the first route request packet it receives from each route discovery. When the route request reached the target, the destination returns a route reply to the node from which it heard that route request. Each node hearing a route reply forwards the reply to the node from which it heard the request. In figure 2-2, when a node B determines that its next-hop to D is unreachable, it broadcasts a signed route error message indicating that its next hop to D is unreachable. Each node using B as a next-hop for D rebroadcasts this route error but does not re-sign it. Because ARAN uses public-key cryptography for authentication, it is particularly vulnerable to DoS attacks based on flooding the network with bogus control packets for which signature verifications are required. As long as a node can’t verify signatures at line speed, an attacker can force the node to discard some fraction of the control packets it receives.

S→* : $(ROUTE\_REQUEST, D, cert_s, N, t)_{k_{s^-}}$

A→* : $((ROUTE\_REQUEST, D, cert_s, N, t)_{k_{s^-}})_{k_{A^-}}, cert_A$

B→* : $((ROUTE\_REQUEST, D, cert_s, N, t)_{k_{s^-}})_{k_{B^-}}, cert_B$

C→* : $((ROUTE\_REQUEST, D, cert_s, N, t)_{k_{s^-}})_{k_{C^-}}, cert_C$

D→C : $(ROUTE\_REPLY, S, cert_D, N, t)_{k_D}$

C→B : $((ROUTE\_REPLY, S, cert_D, N, t)_{k_D})_{k_{C^-}}, cert_C$

B→A : $((ROUTE\_REPLY, S, cert_D, N, t)_{k_D})_{k_{B^-}}, cert_B$

A→S : $((ROUTE\_REPLY, S, cert_D, N, t)_{k_D})_{k_{A^-}}, cert_A$

Figure 2-1: Route discovery in ARAN

B→A : $(ROUTE\_ERROR, S, D, cert_B, N, t)_{k_{B^-}}$

A→S : $(ROUTE\_ERROR, S, D, cert_B, N, t)_{k_{B^-}}$

Figure 2-2: Route maintenance in ARAN

2.2. SEAD

SEAD is robust against multiple uncoordinated attackers creating incorrect routing state in any other node, in spite of active attackers or compromised nodes in the network. To support use of SEAD with nodes of limited CPU processing capability, and to guard against DoS attacks in which an attacker attempts to cause other nodes to consume excess network bandwidth or processing time, we use efficient on-way hash function and don’t use asymmetric cryptographic operations in the protocol.

Distance-Vector Routing In distance-vector routing, each router maintains a routing table listing all possible destinations within the network. Each entry in a node's routing table contains the identity of some destination, the node's shortest known distance of the destination, and the identity of the node's neighbor router that is the first hop on the shortest route to the destination. To maintain the routing tables, each node periodically broadcasts a routing update containing the information from its own routing table. Each node updates its own table using the updates it hears so that its route for each destination uses as a next hop the neighbor that advertised the smallest metric in its update for that destination. It is a Bellmen-Ford algorithm so that the each routing path is minimum distance to the destination.

Hash Chains A one-way hash chain is built on a one-way hash function. A one-way hash function H maps an input of any length to a fixed-length bit string. Thus, $H:\{0,1\}^* \to \{0,1\}^\rho$ where ρ is then length in bits of the hash function's output. To create a one-way hash chain, a node chooses a random $x \in \{0,1\}^\rho$ and computes the list of values $h_0, h_1, h_2, ..., h_n$ where $h_0 = x$, and $h_i = H(h_{i-1})$. The node initially generates the elements of its hash chain using this recurrence, in order of increasing subscript *i*. Given an existing authenticated element of a one-way hash chain, we can verify elements later in the sequence of use within the chain. For example, given an authenticated h_i value, a node can authenticate h_{i-3} by computing $H(H(H(h_{i-3})))$ and verifying that the resulting value equals h_i.

Merkle Hash Trees Merkle hash tree [18] is a mechanism for computing a single cryptographically secure hash digest over a set of data elements. Merkle hash tree is a binary hash tree and in many signature amortization schemes, Merkle hash tree is build on top of the packets' hash values. The internal nodes are recursively defined as hash values which are produced by hashing the concatenation of its two children.

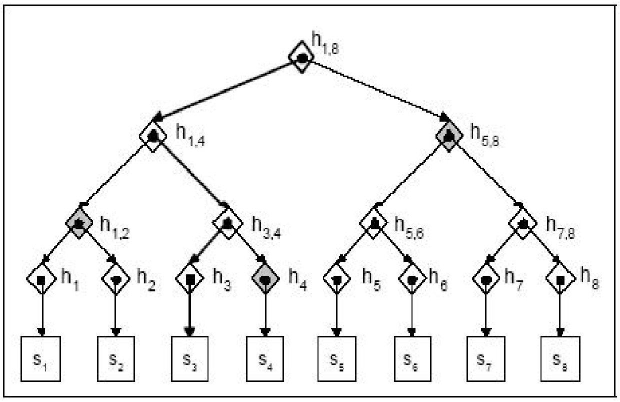

Figure 2-3: Merkle Hash Tree

Figure 2-3 shows a Merkle hash tree and each S_i indicates a packet. Each leaf node h_i is calculated by hashing the corresponding S_i, and each internal nodes $h_{i,j}$ means the hash results of the concatenation of h_i and h_j. The verification sequence of a Merkle hash tree for a leaf node indicates the hash values of the sibling nodes on the path from the leaf node to root. With a leaf node and its verification sequence, the root hash value of the tree can be retrieved. For instance, in Figure 2-3, the verification sequence of packet s_3 can be represented as *(h_4, $h_{1,2}$, $h_{5,8}$)*.

Authenticating Routing Updates Each node in SEAD uses a specific single next element from its hash chain in each routing update that it sends about itself. Based on this initial element, the one-way hash chain conceptually provides authentication for the metric's lower bound in other routing updates for this destination; the authentication provides only a lower on the metric. The method SEAD uses to authenticate an entry in routing update uses the sequence number in that entry to determine a contiguous group of m elements from that destination node's hash chain, one element of which must be used to authenticate that routing updates. Specifically, if a node's hash chain is the sequence of values $h_0, h_1, h_2, ..., h_n$ and n is divisible by m, then for a sequence number i in some routing update entry, let $k = \frac{n}{m} - i$. An element from the group of elements $h_{km}, h_{km-1}, h_{km-2}, ..., h_{km+m-1}$ from this hash chain is used to authenticate the entry; if the metric value for this entry is j, then the value h_{km-j} is used here to authenticate the routing update entry for that sequence number. Nodes receiving any routing update easily can authenticate each entry in the update, given any earlier authentic hash element from the same hash chain.

2.3. Ariadne

The full title of Ariadne is A Secure On-Demand Routing Protocol for Ad Hoc Networks. Ariadne achieves some security requirement by Tesla [28], an efficient broadcast authentication scheme that requires loosy time synchronization. Ariadne's basic idea is based on DSR: it discovers routes on-demand through route discovery and uses them to source route data packets to their destinations. Each forwarding node helps by performing route maintenance to discover problems with each selected route.

Ariadne Route Discovery The first setage of Ariadne route discovery is letting the target verify the authenticity of the ROUTE REQUEST. The initiator first includes a message authentication code computed with key pre-shared with destination node over a timestamp. The target can verify the route request's authenticity and freshness using the shared key. The initiator wants to authenticate each individual node in the node list of ROUTE REPLY. A second requirement is that the target can authenticate each node in the node list of ROUTE REQUEST so that it will return a ROUTE REPLY only along paths that contain legitimate node. Each hop authenticates the new information in the REQUEST using its current Tesla key. The Tesla security condition is verified at the target, and the target includes a MAC in the REPLY to certify that the security condition was met.

The second stage of Ariadne route discovery is per-hop hashing. Authenticating data in routing message isn't sufficient. We use one-way hash functions to verify that no hop was omitted in a REQUEST. To change or remove a previous hop, an attacker must either hear a REQUEST without that node listed or must be able to invert the one-way has function.

Ariadne Route maintenance A node forwarding a packet to the next hop along the source route returns a ROUTE ERROR to the packet's original sender if it is unable to deliver the packet to the next hop. An attacker may also send a forge ERRORS to influence the original route, so preventing unauthorized node from sending ERRORS, we require that the sender authenticate an ERROR. Each node on the return path to the source forwards the ERROR. If the authentication is delayed, each node that will be able to authenticate the ERROR buffers it until it can be authenticated by Tesla.

We have already described some of the outstanding prior works on secure routing above, but all of them have a common drawback that may cause the network disconnected in a short time. The problem comes from the shortest path selecting property which leads to unbalanced path load on each node. To conquer this problem, we propose a new secure routing scheme, called Energy-Balancing Secure Routing. We introduce energy-balancing secure routing scheme in detail in the next section.

3. Proposed scheme

The potential problem in current secure routing protocols is that they tend to find the lowest energy consumption paths for each route. However, the energy depletion always converges on some of the nodes which are on the best paths, and this shortens the total time of overall network connectivity.

To eliminate this problem, we propose a new secure routing scheme called Energy-Balancing Secure Routing. The basic idea of proposed scheme is that the base station can gather node information on the routing path from each route, and according to this information, base station can find nodes that are heavy loaded, called bottleneck node. Our goal is to tune the load of each bottleneck node to average in order to keep the network connected for longer time.

For the requirements for sensor network security, we combine some of the secure building blocks in SPINS [19] and some of the key establishment mechanism in LEAP [20] in the proposed scheme.

We assume:

- Each node knows its own data generating rate.
- Communication between two parties is bidirectional.
- Every node are stationary after deployment, (or move slowly enough for their neighbors to be aware of the changes about the moving node)
- The initial energy of each node is equal.
- There exists a lower bond on the time T_{min} that is necessary for an adversary to compromise a sensor node, and that the time T_{est} for a newly deployed sensor node to discover its immediate neighbor is smaller than T_{min}. This assumption follows that in LEAP.

Notation	description
i	Principal, such as communicating nodes
j	Principal, such as communicating nodes
K_i^m	Individual node key of node i, shared between node i and base station
K_{ij}	Pairwise shared key, only shared between node i and node j
f	A family of pseudo-random function [21]
K_s^m	Base key, only store in base station used to derive individual node key for each node
K_I	Initial key, uses to derive master key of each node
K_i	Master key of node i, which is used for constructing pairwise shared key
K_{ij}^E	Message encryption key, shared between node i and node j
K_{ij}^{MAC}	MAC key, shared between node i and node j
K_i^{mE}	Message encryption key, shared between node i and base station
K_i^{mMac}	MAC key, shared between node i and base station
$Nonce_i$	A nonce generated by node i
$MAC_K(M)$	Message authentication code of message M ,with MAC key K
$E_{K,IV}\{M\}$	Encryption of message M, with encryption key K, and the initialization vector IV which is used in counter mode [22][23][24] for encryption

Table 3-1 Notations used in this thesis

Table 3-1 shows the notation to describe protocols and cryptographic operations in this thesis.

Our proposed scheme is divided into four components: key establish phase, bottleneck node finding mechanism, load balance propagation phase, and secure data transmission phase. We describe those components in the following sub-sections.

3.1. Key Establishment Phase

In key establishment phase, we have two kinds of keys to be constructed in each node *i*. One is individual node key K_i^m , another is pairwise shared key K_{ij} shared with its neighbor node *j*.

- Individual Node Key: Every node has a unique key that it only shared with the base station. This key is used for secure communication between a node and the base station.
- Pairwise Shared Key: Every node shares a pairwise key with each of its one-hop neighbors individually. This key is used to secure update message on each two neighbor nodes.

3.1.1. Establishing Individual Node Keys

The individual node key K_i^m for a node *i* is pre-loaded before its deployment, and is generated as follows: $K_i^m = f_{K_s^m}(i)$, where f is a pseudo-random function and K_s^m is a base key that only stores in base station. Note that the base station only needs to store the base key K_s^m in order to save the storage originally kept for each individual node key. When base station needs to use individual node key for node *i*, it computes K_i^m on the fly.

3.1.2. Establishing Pairwise Shared Keys

Before each node is being deployed, each node pre-distribute an initial key K_I .

Each node *i* can derives a master key as follows: $K_i = f_{K_I}(i)$.

When each node *i* is deployed, it broadcast a HELLO message with nonce, and waits for T_{min} which is the minimum time that an adversary needs to compromise a node. Each neighbor node *j* of node *i* who gets the HELLO message replies its node identity with message authentication code (MAC). Figure 3-1 demonstrates how HELLO message and reply message works.

$$\begin{array}{ll} i \rightarrow broadcast: & i, Nonce_i \\ j \rightarrow i: & j, Nonce_j, MAC_{K_j}(Nonce_i, j, Nonce_j) \end{array}$$

Figure 3-1: HELLO message and reply message

Since node *i* can compute K_j from K_I, it is able to verify the reply message sent by node *j*.

When node *i* have verified the identity of its neighbor node *j*, it computes pairwise shared key K_{ij} with node *j* as follows: $K_{ij} = f_{K_j}(i)$. Node j can also compute K_{ij} independently.

After generating the pairwise shared key with all its neighbor node *j*, it delete K_I and all the keys K_j. So no other nodes can compute the pairwise shared key K_{ij} anymore.

Since we each two neighbor nodes share a pairwise shared key, we make preparation for secure communication by this key. As point out in SPINS, a good security design is not to reuse the same cryptography key for different cryptography primitives; this prevents any potential interaction between the primitives that might introduce a weakness. So we derive independent keys for our encryption and MAC operation from pairwise shared key as follows: $K_{ij}^{E} = f_{K_{ij}}(1)$, $K_{ij}^{MAC} = f_{K_{ij}}(2)$ where K_{ij}^{E} is used for encryption and K_{ij}^{MAC} is used for MAC operation. The same, we derive $K_{i}^{mE} = f_{K_{i}^{m}}(1)$ and $K_{i}^{mMac} = f_{K_{i}^{m}}(2)$ where K_{i}^{mE} is used for encryption and K_{i}^{mMAC} is used for MAC operation for secure transmission between each node *i* to base station. We still need two counters shared by the parties (one of each direction of communication) for the block cipher in counter mode. Since the counter values are not secret, we set these two counters as follows: $C_i = Nonce_i$, $C_j = Nonce_j$ where $Nonce_i$ and $Nonce_j$ just used before.

3.2. Initialization Phase

Step1. Initially, each sensor node set its hop count to "infinite"

Step2. In the beginning, base station broadcast a hop count message with data "Hop count=0"

Step3. When a node receives a hop count message, it compares its hop count value to that in hop count message. If hop count value x in the message is smaller than hop count value minus 1, then sets its hop count value to (x+1)

Step4. If hop count value is changed to y in step3, then the node broadcast a hop count message with data "Hop count=y"

At last, each node is aware of its minimum hop count to base station, and knows the hop count of all its neighbors. So, it is able to construct <u>Farer Neighbor Metric</u> and <u>Lower Neighbor Metric</u> in the next phase.

3.3. Bottleneck Node Finding Mechanism

Step1. First of all, every node contains one variable (Load Count ($\boldsymbol{L_i}$)) and two metrics (Farer Neighbor Metric, Lower Neighbor Metric). Initially, each node set its Load Count ($\boldsymbol{L_i}$) according to its data generating rate, and in Farer Neighbor Metric, it sets Path Load Count to 0 and sets IsBalanceable to False for those nodes that hop count number is grater than current node. In Lower Neighbor Metric, each column is set to be $\left(\frac{L_i}{Number\_of\_Lower\_Nodes}\right)$ which represents the Arrange Load of those nodes that hop count number is lower than current node. Figure 3-2 demonstrates variables maintained in a node, and figure 3-3 shows a Farer Neighbor Metric, and figure 3-4 shows a Lower Neighbor Metric.

Current ID	Load Count
55	2

Figure 3-2: Node variable

Neighbor ID	Path Load Count	IsBalanceable
6	0	False
285	0	False
60	0	False

Figure 3-3: Farer Neighbor Metric

Neighbor ID	Arrange Load
57	1
74	1

Figure 3-4: Lower Neighbor Metric

Step2. Every node that has no farer neighbor is the starting point in this step. When node *i* is going to send a message through node *j* where node *j* are lower neighbors of node *i*, it computes Path Load Count with $(L_i \bullet P_{ij})$ and adds Route Path, Path Load Count, Max Load Count, and Max Load Node ID to the message, where Max Load Count is set to Load Count of node *i* ***(L_i)***, and Max Load Node ID is set to node ID of node *i*. IsBalanceable is set true if current node has more than one lower neighbors, otherwise set false.

When route message (RM) is ready, then it is sent in secure manner as showed below.

<table>
<tr><td colspan="2">Route Path</td><td>Path Load Count</td></tr>
<tr><td>Max Load Count</td><td>Max Load Node ID</td><td>IsBalanceable</td></tr>
</table>

Figure 3-5: Format of an Route Message (RM)

$$\boxed{i \rightarrow j : \quad E_{K_{ij}^{E}, C_i}\{RM\}, MAC_{K_{ij}^{MAC}}(C_i, E_{K_{ij}^{E}, C_i}\{RM\})}$$

Figure 3-6: Secure transmission of Route Message

Step3. In this step, every intermediate node must receive all the route messages that all its farer neighbors send, and then progresses. For each intermediate node *j*, it gets the Path Load Count value within the message, and updates the value in its Farer Neighbor Metric. If Farer Neighbor Metric changes, node *j* has to recount its Load Count which is the summation of all Path Load Count in Farer Neighbor Metric ($Load\_Count = \sum_{each\_node\_in\_Farer\_neighbor\_metric} Path\_Load\_Count$). If current Load Count is grater than Max Load Count in route message, then changes Max Load Count and Max Load Node ID to current node. If IsBalanceable is false and current node has more than one lower neighbor, then change it to true. At last, node *j* adds its node ID to Route Path and changes Path Load Count to $(L_j \bullet P_{jk})$ where *k* is all of its lower neighbors and then sends this message to node *k*.

Also, the route message is sent in secure manner as showed below.

$$\boxed{j \rightarrow k: \quad E_{K_{jk}^{E},C_j}\{RM\}, MAC_{K_{jk}^{MAC}}(C_j, E_{K_{jk}^{E},C_j}\{RM\})}$$

Figure 3-7: Secure transmission on intermediate node

Step4. In this step, base station will get all routes, and every route contains a bottleneck node which has the maximum load on this route. So when a route message reaches base station, base station saves the Max Load Count, Max Load Node ID and Route Path to Bottleneck List which stored at base station.

BTNK ID	Load	Route Path
3	9	3,1,4,5,8
35	15	35,44,58,90,23
366	10	10,366,289,411
41	23	25,31,566,2,98
…	…	…

Figure 3-8: Bottleneck List

After this phase, base station knows the overall load of the whole network and those nodes that are most heavy loaded in each routing. Our goal is to balance the load of bottleneck node in order to make the energy usage of these nodes to be equal.

3.4. Load Balance Propagation Phase

Step1. Base station counts expected load of each bottleneck node followed by this formulation: $\left(\frac{overall\_load\_of\_network}{number\_of\_bottleneck\_node}\right)$, and then sends the expected load to each bottleneck node by piggybacking.

Step2. When a bottleneck node is informed by base station with expected load, it compares its current load to the expected load, if current load is larger than expected load, then progresses Step3. If current load is smaller than expected load, then progresses Step4. Otherwise this node is load balanced.

Step3. In this step, because load of current node is heavier than expected load, we need to reduce the load of this node. We inform the farer neighbors whose IsBalanceable flag is true by Reduce Message. Reduce Value in the Reduce Message is set according to the percentage of its load in Farer Neighbor Metric followed by formulation:

$$\mathrm{Re}\,duce\_Value\_of\_Neighbor\_Node_i =$$

$$\left((Current\_Load - Expected\_Load) \bullet \frac{Path\_Load\_Count_i}{\sum_{Node_j} Path\_Load\_Count_j} \right)$$

where $Node_j$ are nodes with IsBalanceable flag is true in Farer Neighbor Metric.

Reduce Message is sent to next node with secure encryption protocol just like showed below.

Node ID:10	Load: 60	Expected Load: 50
Neighbor ID	Path Load Count	IsBalanceable
6	10	False
285	20	True
60	30	True

Figure 3-9: Node variables and Far Neighbor Metric

Node ID:10	Neighbor ID:285	Reduce Value:4

Figure 3-10: Reduce Message to Neighbor 285

Node ID:10	Neighbor ID:60	Reduce Value:6

Figure 3-11: Reduce Message to Neighbor 60

$$i \rightarrow j: \quad E_{K_{ij}^{E},C_i}\{RM\}, MAC_{K_{ij}^{MAC}}(C_i, E_{K_{ij}^{E},C_i}\{RM\})$$

Figure 3-12 Secure transmission of reduce message

Step4. In this step, because load of current node is lighter than expected load, we want to increase the load of this node. We inform the farer neighbors whose IsBalanceable flag is true by Increase Message. Increase Value in the Increase Message is set according to the percentage of its load in Farer Neighbor Metric followed by formulation:

$$Increase\_Value\_of\_Neighbor\_Node_i =$$

$$\left((Expected\_Load - Current\_Load) \bullet \frac{1/Path\_Load\_Count_i}{\sum_{Node_j} \left(1/Path\_Load\_Count_j \right)} \right)$$

where $Node_j$ are nodes with IsBalanceable flag is true in Farer Neighbor Metric.

The same, Increase Message is sent to next node with secure encryption protocol just like showed below.

Node ID:10	Load: 60	Expected Load: 70
Neighbor ID	Path Load Count	IsBalanceable
6	10	False
285	20	True
60	30	True

Figure 3-13: Node variables and Far Neighbor Metric

Node ID:10	Neighbor ID:285	Increase Value:6

Figure 3-14: Increase Message(IM) to Neighbor 285

Node ID:10	Neighbor ID:60	Increase Value:4

Figure 3-15: Increase Message(IM) to Neighbor 60

$$\boxed{i \rightarrow j: \quad E_{K_{ij}^{E},C_i}\{IM\}, MAC_{K_{ij}^{MAC}}(C_i, E_{K_{ij}^{E},C_i}\{IM\})}$$

Figure 3-16: Secure transmission of reduce message

Step5. Every node that receives Reduce Message and Increase Message is able to balance load of lower neighbors. Node can change Arrange Load according to Reduce Value and Increase Value in the messages. Once sum of Reduce Value and sum of Increase Value are not equal, the node needs to send Reduce Message or Increase Message just like what we have done in step3 and step4, otherwise load balance is done.

Step6. Once load balancing message (Reduce Message and Increase Message) can not be able to propagate anymore, we can restart doing bottleneck finding mechanism for more balanced network or just stop to save the energy cost when applying these mechanisms according to the balance level that user wants.

3.5. Secure Data Transmission Phase

After bottleneck finding mechanism and load balance propagation phase is applied, Arrange Load in Lower Neighbor Metric of each node is changed for the purpose of balancing energy consumption of whole network. In this phase, if a node generates a message to base station, it chooses its next hop node according to the probability of Arrange Load. Probability of each lower neighbor being choose is addressed by the following formulation:

$$P_{ij} = \frac{Arrang\_Load_j}{\sum_k Arrang\_Load_k}$$

where i is current node and k is each node in Lower Neighbor Metric of node i

When next hop is chosen by the probability, then the message M it wants to send to base station will be transmitted through secure encryption protocol. We will show the secure encryption protocol below.

Neighbor ID	Arrange Load
56	10
55	15
54	25

Figure 3-17: Load in Lower Neighbor Metric

$P_{10,56}$ =20%
$P_{10,55}$ =30%
$P_{10,54}$ =50%

Figure 3-18: Probability of choosing next hop

Message M with data D is going to send to base station by node *i* through node *j*:

$$i \to j : E_{K_{ij}^{E}, C_i}\{M\}, MAC_{K_{ij}^{MAC}}(C_i, E_{K_{ij}^{E}, C_i}\{M\})$$

where

$$M : E_{K_i^{mE}}\{D, Nonce_i\}, MAC_{K_i^{mMAC}}(E_{K_i^{mE}}\{D, Nonce_i\})$$

4. Evaluation

In this section, we use some different kinds of network as our experimentation environment. We will make some evaluations to show the total time of overall network connectivity of proposed scheme under different network conditions. For comparison, we choose some different kinds of secure routing schemes that trying to find minimum energy cost paths and an energy aware routing scheme without security transmission. As point out in [25], performance of AODV is similar to that of DSR, but AODV has more expensive routing overhead than DSR, so we cast aside AODV based secure routing for comparison. For secure routing scheme, SEAD is selected to be compared because of its DSDV routing based property and another chosen scheme is Ariadne which is based on DSR. For traditional routing scheme, we also compare to Energy Aware Routing [16] proposed by Rahul et al which is going to balance energy usage when selecting next hop.

4.1. Simulation Setup

We implement SEAD, Ariadne, Energy Aware Routing, and Energy-Balancing Secure Routing using Wireless Sensor Network Simulator, which is a JAVA based simulator. According to the spec of Crossbow MICAz [26], power consumption of radio-frequency transceiver and micro-controller computation are listed in Table 4.1.1.

According to the spec of TinyOS , the embedded operating system used on Micaz, we set a data message packet to be 30 bytes. Since the size of encrypted message is the same as the size of the plaintext, there is no packet overhead within encrypting operation. MAC operation adds 8 bytes to a message, but MAC gives up integrity guarantees, so we don’t need extra 2 bytes of CRC. Thus, there is only 6 bytes packet overhead in MAC operation. We set nonce to be 2 bytes for a large enough value range. Table 4.1.2 shows packet overhead used in proposed scheme.

Component	Energy Dissipation
Processor in Active Mode	24 mW
Processor in Sleep Mode	5 μW
Transceiver in Receive Mode	59.1 mW
Transceiver in Transmit Mode	52.2 mW
Transceiver in Idle Mode	60 μW
Transceiver in Sleep Mode	3 μW

Table 4.1.1

Security operation/ component	Packet overhead
Symmetric-Key Encryption	0 bytes
Message Authentication Code	6 bytes
Nonce	2 bytes

Table 4.1.2

MICAz uses two AA batteries for power supply. If we put two 1.5V 2000mAh AA batteries on MICAz, then the total energy of a MICAz node is $2.16\times10^{4}J$. So initially, we put 500 sensor nodes in a 700m x 700m square filed with base station settled at the central of the field, and each node carries $2.16\times10^{4}J$ energy. Deployment of sensor nodes follows normal distribution with mean value set at base station and standard deviation is set 140. The communication range of each node is 70 meter.

4.2. Evaluation on normally distributed network

In this sub-section, we are going to evaluate the performance in the network which the node deployment is followed by normal distribution.

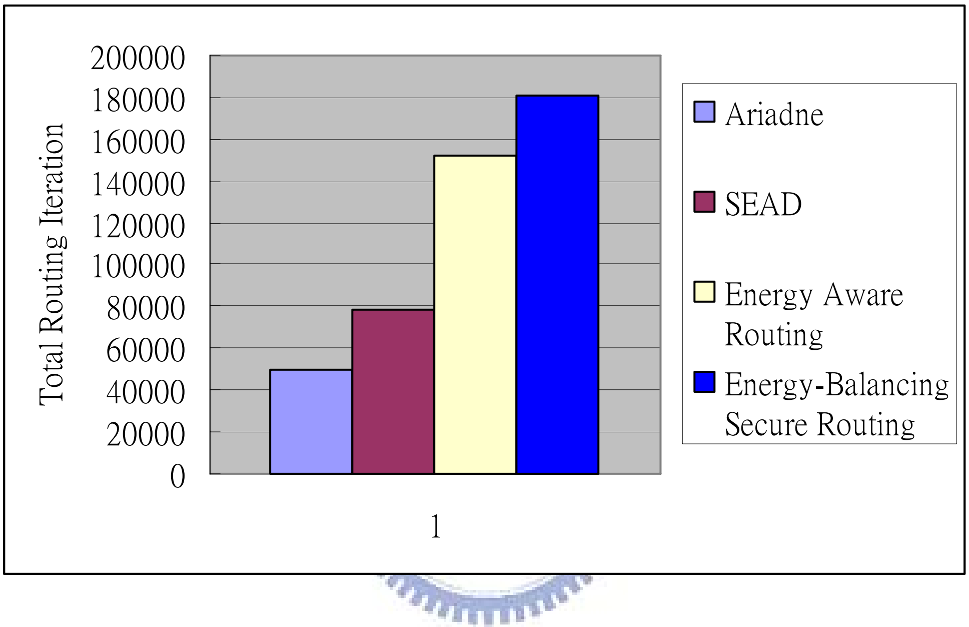

Figure 4-1: Evaluation on initial conditions

Figure 4-1 shows the evaluation result on initial conditions. We from the result, we can see that the network lifetime of Energy-Balancing Secure Routing is about 4 times than that of Ariadne, and is about 2 to 3 times than that of SEAD. The comparison between Energy-Balancing Secure Routing and Energy Aware Routing is shown that the proposed scheme increases network lifetime about 10 to 20 percent.

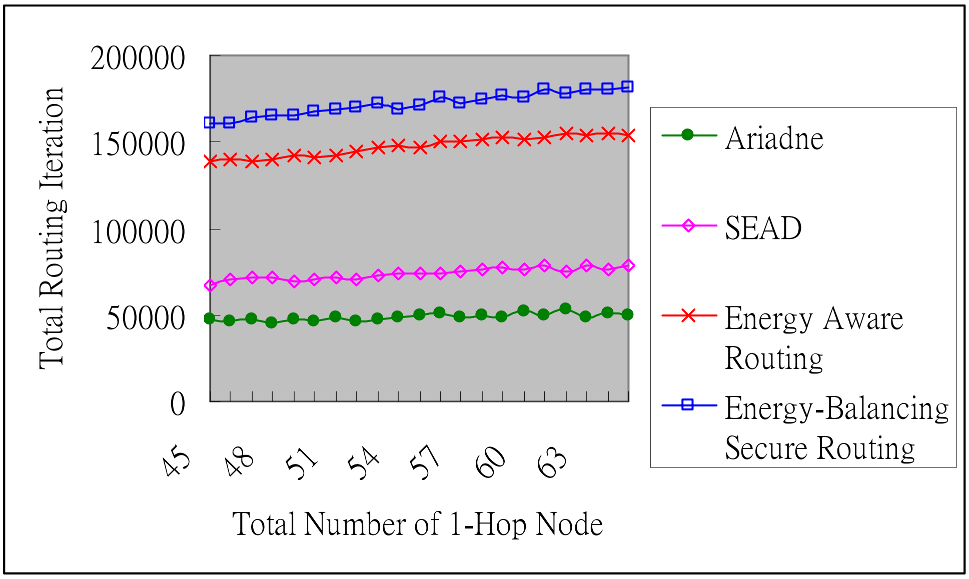

Figure 4-2: Evaluation on different number of 1-hop node

We evaluate the performance when the number of 1-hop nodes increase. Here, 1-hop nodes means for the nodes with its minimum hop count to base station is 1, on the other hand, these are the nodes that are neighbors of base station. Figure 4-2 shows that when the number of 1-hop node increase, the proposed scheme can enlarge more total network lifetime. That's because in the proposed scheme, the more 1-hop node we have, the more node we can perform energy balance propagation.

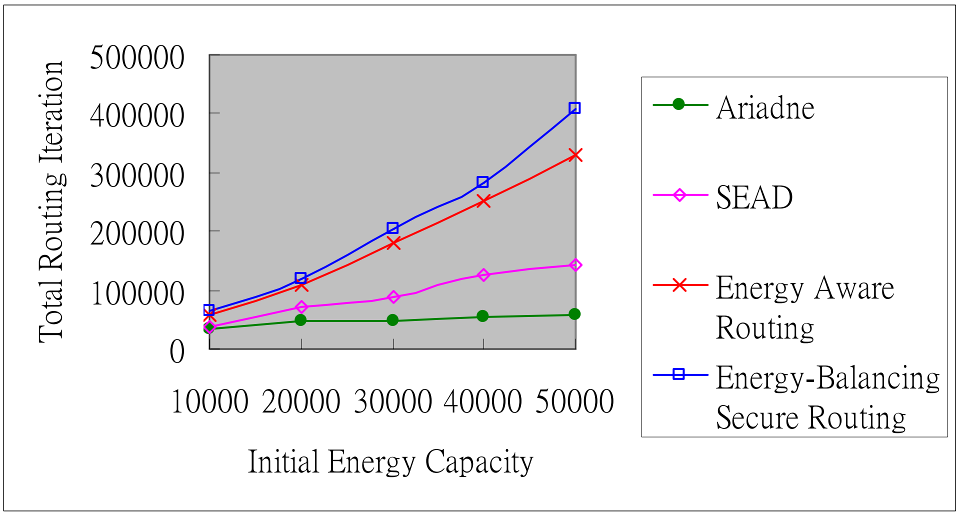

Figure 4: Evaluation on different initial energy capacity

Figure 4-3 shows the evaluation result when the initial energy capacity is different. We observe that as the initial energy capacity increase, the enhancement of total network lifetime is more significant. That's because the more energy that each node have, the more energy can be balanced. Although the present energy capacity of a MICAz node is about 30000 units in this figure, the evolution of sensor node will be design for larger energy capacity of a node. The propose scheme may be more flexible when the design of sensor node to be with more energy capacity.

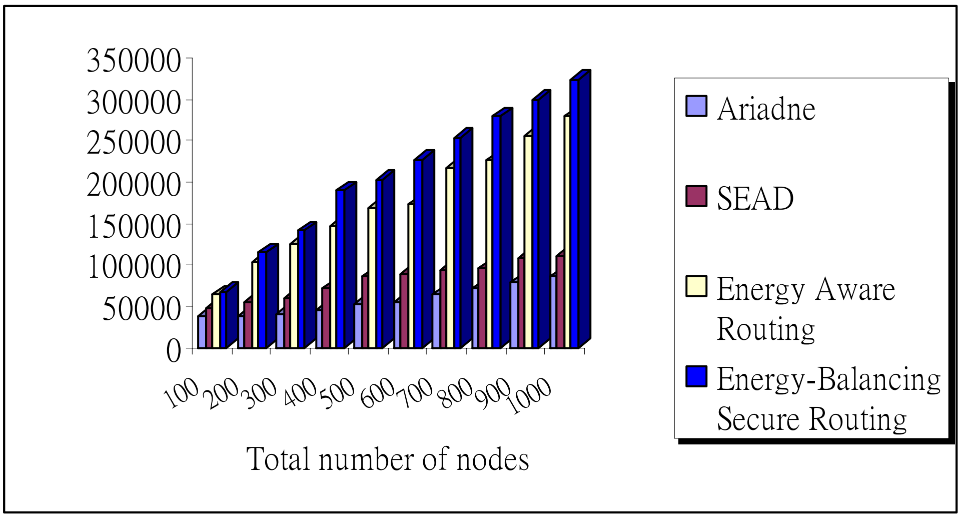

Figure 5: Evaluation on total number of nodes

Figure 4-4 illustrates the performance for a variable number of nodes. The results show that the total network lifetime significantly increases as the total number of nodes increase. Thus, our proposed scheme performs well in large scale networks. Since real world applications utilizing wireless sensor networks may employ thousands of sensor nodes, the evaluation demonstrates that Energy-Balancing Secure Routing is suitable for this type of network deployment.

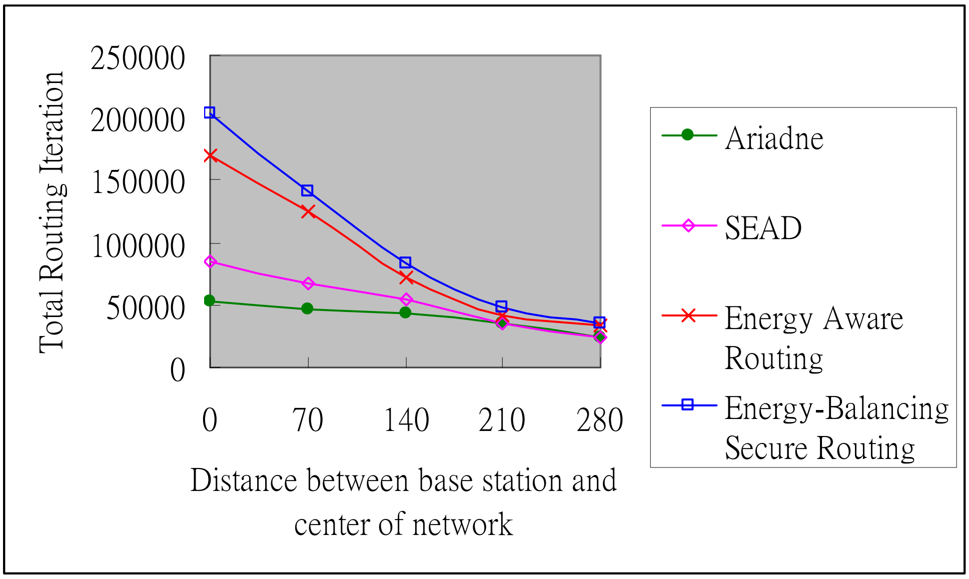

Figure 6: Evaluation on distance between base station and central of the network

Figure 4-5 demonstrates the situation on variety distance between base station and central of the network. The result shows that the total routing iterations decrease as distance between base station and central of the network drop off. That is because when base station far away from central of network, 1-hop neighbor will reduce significantly, and it is harder to balance the energy consumption caused by routing. However, the evaluation shows that Energy-Balancing Secure Routing still perform better than other scheme because of no periodically broadcasting cost in the proposed scheme.

The evaluation shows that Energy-Balancing Secure Routing can prolong total time of overall network connectivity in a multiple times than present secure routing protocols, and is adaptable to the network environment with more 1-hop node. In the future, energy capacity of each sensor node will be design larger, and then the proposed scheme will perform better.

5. Security Analysis

In this section, we analyze the security in the proposed scheme. We first list the requirements for wireless sensor networks, and then discuss the protection of the requirements.

5.1. Security Requirements

We list the security properties required by wireless sensor network, and shows how they directly applicable in a typical wireless sensor network.

5.1.1. Semantic Security

In sensor networks, due to wireless communication property, eavesdropper can always get the transmitting messages on the air. [27] mentions that eavesdropper may be able to infer the message content from the encrypted messages if it sees multiple encryptions of the same plaintext. Semantic security is a strong security property that prevents eavesdroppers from getting information about the plaintext from multiple encrypted messages.

5.1.2. Data Confidentiality

A wireless sensor network should not leak sensor readings to neighboring networks. In many applications, nodes communicate highly sensitive data. The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, hence achieving data confidentiality.

5.1.3. Data Authentication

Message authentication is important for many applications in wireless sensor networks. Since an adversary can easily inject messages, the receiver needs to ensure that data used originates from a trusted source. Informally, data authentication allows a receiver to verify that the data really was sent by the claimed sender.

In two-party communication case, data authentication can be achieved through a purely symmetric mechanism: the sender and the receiver shared a secret key to compute a message authentication code (MAC) of all communicated data. When a message with a correct MAC arrives, the receiver knows that it must have been sent by the sender.

5.1.4. Data Integrity

In network communication, data integrity ensures the receiver that the received data is not altered in transit by an adversary. Data integrity can be achieved through data authentication, which is a stronger security property.

5.1.5. Data Freshness

Wireless sensor networks send message over time, so it is not enough to guarantee confidentiality and authentication; we must also ensure each message is fresh. Informally, data freshness implies that the data is recent, and it ensures that no adversary replayed old message.

5.2. Analysis

In this section, we are going to show how the proposed scheme provides semantic security, data authentication, data confidentiality, data integrity, replay protection and data freshness.

5.2.1. Semantic Security

A basic technique to achieve semantic security is randomization: Before encrypting the message with a chaining encryption function, the sender precedes the message with a random bit string. This prevents the attacker from inferring the plaintext-ciphertext pairs encrypted with the same key. However, sending the randomized data over a wireless channel requires more energy.

Alternatively, the proposed scheme constructs another cryptographic mechanism that achieves semantic security without additional transmission overhead. We use a counter pair shared by two communication parties for the block cipher in counter mode. Since the counter value is incremented after each message transmission, the same message is encrypted differently each time. Hence, the eavesdropper is not able to derive knowledge from all the encrypted messages flowing over the network.

5.2.2. Data Authentication and Data Integrity

In the proposed scheme, each two communication parties shares a pair of secret key for MAC operation. When receiver receives a message, it can use the MAC key pair of the claimed sender to verify the MAC is correct or not. Because on other nodes contain this MAC key pair, the correctness of the MAC verification can prove that the message is sent from the claimed sender.

5.2.3. Data Confidential

In the proposed scheme, each two communication parties shares a pair of secret key for encryption. We encrypt message with the secret key while being transmit. No other node can know the encrypted message because the lack of secret key, hence achieving data confidentiality

6. Conclusion

Present secure routing schemes have the same fatal wound that they may drain some of the nodes in the network in a very short time, and that may lead to the disconnected of the network. In this paper, we proposed Energy-Balancing Secure Routing scheme that can prolong the total time of overall network connectivity. In the proposed scheme, we first setup secrete keys shared with the two communication parties for symmetric key cryptography primitives. Then we balance the routing selection probability of each node to prevent any node from being heavy loaded. Finally, we setup the secure communication channel from each node to base station on the balance loaded network.

The evaluation shows that the proposed scheme can prolong the total time of overall network connectivity in a multiple times than presents secure routing scheme, and perform well in the network environment with more 1-hop node. The security analysis shows that energy-balancing secure routing scheme achieves semantic security, which prevents eavesdropper infers the message content from the encrypted messages if it sees multiple encryptions of the same plaintext.

References

[1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. Atlanta, Georgia, USA: ACM Press, 2002, pp. 88-97.

[2] Z. Butler, P. Corke, R. Peterson, and D. Rus, "Virtual fences for controlling cows," presented at *Proceedings of 2004 IEEE International Conference on Robotics and Automation (ICRA '04)*, 2004.

[3] C. Lin, C. Federspiel, and D. Auslander, "Multi-Sensor Single Actuator Control of HVAC Systems," presented at *International Conference for Enhanced Building Operations*, 2002.

[4] I. F. Akyildiz, W. S. Sankarasubramaniam, and E. Y. Cayirci, "A survey on sensor networks," in *Communications Magazine, IEEE*, 2002.

[5] M. Tubaishat and S. Madria, "Sensor networks: an overview," in *Potentials, IEEE*, 2003.

[6] S. S. Intille, "Designing a home of the future," *Pervasive Computing, IEEE*, 2002.

[7] D. H. Goldberg, A. G. Andreou, P. Julien P. O. Pouliquen, L. Riddle, and R. Rosasco, "A wake-up detector for an acoustic surveillance sensor network: algorithm and VLSI implementation," in *Proceedings of the third international symposium on Information processing in sensor networks*. Berkeley, California, USA: ACM Press, 2004, pp. 134-141

[8] L. Schwiebert, S. K. S. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," in *Proceedings of the 7th annual international conference on Mobile computing and networking*. Rome, Italy: ACM Press, 2001,

pp. 151-165.

[9] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," presented at *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*. Proceedings. IEEE, 2000.

[10] Y.-C. Hu, D. B. Johnson, and A. Perrig, "SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks," presented at *Mobile Computing Systems and Applications, 2002*. Proceedings Fourth IEEE Workshop on, 2002.

[11] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, pp. 234-244, 1994.

[12] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," presented at *Second IEEE Workshop on Mobile Computer Systems and Applications*, 1999.

[13] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad hoc networks," presented at *Network Protocols, 2002. Proceedings*. 10th IEEE International Conference on, 2002.

[14] Yih-Chun Hu, Adrian Perrig, and David B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," *Wireless Networks*, vol. Volume 11, pp. Pages 21 - 38, 2005.

[15] M. G. Zapata and N. Asokan, "Securing ad hoc routing protocols," in *Proceedings of the 3rd ACM workshop on Wireless security*. Atlanta, GA, USA: ACM Press, 2002, pp. 1-10.

[16] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," presented at *Proceedings of Wireless Communications and Networking Conference, 2002*. WCNC2002. 2002 IEEE, 2002

[17] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: the dynamic source routing

protocol for multihop wireless ad hoc networks " in *Ad hoc networking* Addison-Wesley Longman Publishing Co., Inc., 2001 pp. 139-172

[18] R. Merkle. Protocols for public key cryptosystems. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 122–134, Apr. 1980.

[19] Adrian Perrig, Robert Szewczyk, J.D. Tygar, Victor Wen, and David E. Culler, "SPINS: Security Protocols for Sensor Networks," *Wireless Networks*, vol. Volume 8, pp. Pages 521 - 534, 2002.

[20] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *Proceedings of the 10th ACM conference on Computer and communications security*. Washington D.C., USA: ACM Press, 2003, pp. 62-72.

[21] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *J. ACM*, vol. 33, pp. 792-807, 1986.

[22] Mihir Bellare, Anand Desai, Eron Jokipii, and P. Rogaway, "A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES modes of Operation," presented at Symposium on Foundations of Computer Science (FOCS), 1997.

[23] W. Diffie and M. E. Hellman, "Privacy and authentication: An introduction to cryptography," *Proceedings of the IEEE*, vol. 67, pp. 397-427, 1979

[24] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, “Handbook of Applied Cryptography,” *CRC Press*, 1997.

[25] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*. Dallas, Texas, United States: ACM Press, 1998, pp. 85-97.

[26] Crossbow Technology, MICAz Datasheet, http://www.xbow.com/Products /Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf

[27] S. Goldwasser and S. Micali, Probabilistic encryption, *Journal of Computer Security*

[28] Adrian Perrig, J.D. Tygar, Dawn Song, and Ran Canetti, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," presented at 2000 IEEE Symposium on Security and Privacy (S&P 2000), 2000.

[29] I. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[30] H. Krawczyk. Distributed fingerprints and secure information dispersal. In 13th ACM Symposium on Principles of Distributed Computing, pages 207–218. ACM, 1993.

[31] R. Merkle. Protocols for public key cryptosystems. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 122–134, Apr. 1980.

[32] P. Golle and N. Modadugu. Authenticating streamed data in the presence of random packet loss. In *Proceedings of the Symposium on Network and Distributed Systems Security (NDSS 2001)*, pages 13–22. Internet Society, Feb. 2001.

[33] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. Efficient authentication and signature of multicast streams over lossy channels. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 56–73, May 2000.

[34] D. Song, D. Zuckerman, and J. D. Tygar. Expander graphs for digital stream authentication and robust overlay networks. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 258–270, May 2002.