A Secure Scheme for Power Exhausting Attacks in Wireless Sensor Networks

Ching-Tsung Hsueh, Chih-Yu Wen and Yen-Chieh Ouyang*
Department of Electrical Engineering & Graduate Institute of Communication Engineering
National Chung Hsing University
Taichung, Taiwan 40227

* Author to whom correspondence should be addressed; E-Mail: ycouyang@nchu.edu.tw

Abstract— Security and energy efficiency are the most important concerns in wireless sensor networks (WSNs) design. To save the power and extend the lifetime of WSNs, various media access control (MAC) protocols are proposed. Most traditional security solutions can not be applied in the WSNs due to the limitation of power supply. The well-known security mechanisms usually awake the sensor nodes before the sensor nodes can execute the security processes. However, the Denial-of-Sleep attacks can exhaust the energy of sensor nodes and shorten the lifetime of WSNs rapidly. Therefore, the existing designs of MAC protocol are insufficient to protect the WSNs from Denial-of-Sleep attack in MAC layer. The practical design is to simplify the authenticating process in order to enhance the performance of the MAC protocol in countering the power exhausting attacks. This paper proposes a cross-layer design of secure scheme integrating the MAC protocol. The analyses show that the proposed scheme can counter the replay attack and forge attack in an energy-efficient way.

Keywords— wireless sensor networks, energy efficiency, denial-of-sleep, power exhausting attacks, secure scheme.

I. INTRODUCTION

To save energy and extend the lifetime of WSNs, different schemes are researched and proposed [1], [2]. Most of the researchers aim at layer-2 protocol design. In the duty-cycle based WSNs MAC protocols, the sensor nodes are switched between awake/active and sleep state periodically. The sensor nodes are switched into sleep mode after certain idle period [3]-[6]. In the Low Power Listening (LPL) based WSNs MAC protocol, such as B-MAC [3], the receiver wakes up periodically to sense the preamble from the sender and then to receive and process the data. When the sender needs to send data, it sends a long preamble to cover the sleep period to ensure the receiver waking up and sensing. The LPL based MAC protocol is an asynchronous protocol. It decouples the sender and receiver with time synchronization. The X-MAC protocol improves LPL based MAC protocol by replacing the long preamble with shot preambles [6]. Fig. 1 shows the timeline of X-MAC protocol, which allows the receiver to send acknowledgment (ACK) back to the sender as soon as it senses the preamble.

The Denial-of-Sleep is one of the power exhausting attacks of WSNs [7]-[9]. This attack tries to keep the sensor nodes awake to consume more power. In any security mechanism, the sensor nodes must be waked before receiving data and checking security properties. Current layer-2 protocol designs are insufficient to protect a WSN from Denial-of-Sleep attack [7].

Without security mechanism, an anti-node can broadcast a fake preamble frequently. If the receiver can not tell the real preamble and the fake one, the receiver will receive and process the data from the anti-node. Such attack will keep the receiver awake as long as the data transmission sustains, which exhausts the battery of nodes rapidly. Moreover, an anti-node can replay a fake preamble ACK to the sender. Thus, the sender will start to send the data to the anti-node but it will never receive the right data ACK. Similarly, the sender may send data repeatedly and exhausts the battery of node rapidly. As a result, the sender and receiver need mutual authentication schemes to counter such attacks.

In traditional wireless security mechanisms, the transmitting data is encrypted with keyed symmetric or asymmetric encryption algorithm. The wireless sensor networks prefer the symmetric algorithm to avoid the complicated computing and heavy energy consumption. But the encrypted data makes the battery exhaustion even worse under Denial-of-Sleep attack. The anti-node can send the encrypted "garbage" data to receiver. This attack forces the receiver to decrypt the data. Before the receiver identifies that the data is "garbage", the receiver consumes more power to receive and decrypt data. These processes also keep sensor nodes awake longer. An easy and fast mutual authentication scheme is needed to integrate with MAC protocol to counter the encrypted "garbage" data attack.

In this paper, a cross-layer design of secure scheme integrating the MAC protocol, Two-Tier Energy-Efficient
Secure Scheme (TE₂S), is proposed to protect the WSNs from the above attacks. The practical design is to simplify the security process when suffering the power exhausting attacks. The design principles and features of the proposed secure scheme are:

1) Energy conservation
2) Low complexity
3) Mutual authentication
4) Symmetric encryption
5) Dynamic session key generated with challenge text
6) Capability to counter the replay attack and forge attack
7) Integrating the MAC protocol

This paper proposes a two-tier secure transmission scheme. This scheme uses the hash-chain to generate the dynamic session key, which can be used for mutual authentication and the symmetric encryption key. The only computations of dynamic session key are the hash functions, such as MD5 or SHA-1, which are very simple and fast. By integrating with MAC protocol, there is no extra packet compared with the existing MAC designs. The security analysis shows that this scheme can counter the replay attack and forge attack, and the energy analysis shows that this scheme is energy efficient as well.

II. RELATED WORKS

In [10]-[12], a dynamic session key policy (DSKP) was proposed based on a one time password (OTP) system to protect users during the authentication process and session key agreement process. The reason for using OTP is that it varies with sessions where the length is long enough compared with a human-chosen password [13]. Therefore, it is hard to trace and detect. By using the counter indicated hash-chain algorithm, the DSKP is computational cheap. But the synchronized counter of hash-chain algorithm may not be suited to the asynchronous LPL based MAC protocol in WSNs.

The overhead of security algorithms have been well studied on embedded systems [14]-[16]. Several popular algorithms of symmetric encryption and hashing function were evaluated on varied micro-controller units (MCU) in [15]. Based on experimental tests, the clock cycles and execution time were measured for each algorithm and platform. These analytical models can be derived to indicate the computational cost of given embedded architectures on different encryption schemes.

III. NOTATIONS TO BE USED

In order to facilitate and clarify our presentation the following notations are used in the paper.

1. IDₓ: X's identity.
2. Kₛ: session key.
4. Rₛ, Rₓ: random number selected by sender and receiver respectively.
5. h(x): a one-way hash function which x is the input.
6. Eₓ(x): encrypts x by using symmetric algorithm with key K.
7. Dₓ(x): decrypts x by using symmetric algorithm with key K.
8. MACₓ(x): message authentication function with key K, where x is the input message.
9. P_Tx: power of radio in transmit mode.
10. P_Rx: power of radio in receive mode.
11. P_Ri: power of radio in idle mode.
12. P_Rs: power of radio in sleep mode.
13. P_Mc: power of MCU in active mode.
14. P_Mi: power of MCU in sleep mode.
15. T_S: duration of node sleep.
17. T_P: duration of preamble.
18. T_PA: duration of preamble ACK.
19. T_PAL: duration of preamble ACK listening.
20. T_L: duration of listening in steps.
21. T_AC: duration of MCU active and computing in steps.
22. T_TX: duration of radio transmitting in steps.
23. T_RX: duration of radio receiving in steps.

IV. THE SECURE TOPOLOGY FORMATION STAGE

In this stage, the secure adaptive topology control algorithm (SATCA) is involved to form the hierarchical topology in four phases: (I) anti-node detection; (II) cluster formation; (III) key distribution; (IV) key renewal [17].

![Figure 2. Key distribution for WSNs](image)
In phase I, an authenticated broadcasting mechanism is applied to identify the anti-nodes. In phase II, the adaptive distributed topology control algorithm (ADTC) [18] performs the clusterhead selection and the gateway selection to form the clusters. In the phase III, two symmetric keys, a cluster key and a gateway key, are distributed locally under cluster construction. The securities of intra- and inter-cluster communication are established upon the cluster key and the gateway key respectively. In phase IV, the key renewing process revokes the old keys and accomplishes the renewal of the keys. The process of key distribution is shown in Fig. 2.

V. DESIGN PRINCIPLES OF TE2S

After the secure topology formation stage, there is a shared secret key between the valid member nodes and clusterhead of each cluster. A cluster key is a key shared by a clusterhead and all its cluster members, which is mainly used for securing local broadcast messages (e.g. routing control information or sensor messages). Based on the secure cluster topology, a two-tier security scheme is performed to transmit information securely and quickly. This scheme can assist the nodes in deciding to switch into sleep mode or to keep awake as soon as possible. In this work, the X-MAC protocol is involved as the basic architecture of the proposed security scheme [6]. The behavior of packet exchange in the X-MAC protocol is shown in Fig. 3.

A. Tier-1: Session Key Agreement

In Tier-1, a hash-chain is created by using the cluster key $K_c$, which is the shared secret between the valid members and the clusterhead. This hash-chain is used for mutual authentication and symmetric encryption key. A detailed implementation is described as follows (Fig. 4):

Step 1: The sender selects a random number $R_s$ and computes the secure token (i.e. $Token = h(K_c | R_s)$).
Step 2: The sender sends its ID, receiver’s ID, secure token and random number $R_s$ as the preamble.
Step 3: The receiver verifies the secure token. If the token is not valid, the receiver goes back to sleep mode immediately. If the token is valid, then receiver selects a random number $R_r$ and computes the session key $K_s = h(K_c | R_s | R_r)$. The receiver also computes the hash chain $h(K_c)$ and $h(h(K_c))$.
Step 4: The receiver sends the $h(h(K_c))$ and random number $R_r$ as the ACK.
Step 5: The sender computes the session key $K_s = h(K_c | R_s | R_r)$ and the hash chain $h(K_c)$ and $h(h(K_c))$. The sender then verifies the $h(h(K_c))$. If the $h(h(K_c))$ is not valid, the sender will not send the data.

To check the secure token valid, the receiver executes 1 hash function computing and 1 comparing computing. These 2 computations are very simple and fast. If the secure token is not valid, the receiver goes back to sleep mode immediately and discards all the rest processes. To check the receiver is valid, the sender computes and compares the received $h(h(K_c))$. If the $h(h(K_c))$ is not valid, the sender will not send the data. The hash chain $h(K_c)$ and $h(h(K_c))$ are computed for mutual authentication.

Therefore, the sender and receiver reach a dynamic session key agreement with only one random number selection and three hash function computations respectively. This key agreement does not involve any encryption/decryption computing. The random number is the function of timer to make the operation of the random number generator simple and fast.

B. Tier-2: Data Transmission

With the new created dynamic session key $K_s$, the sender can encrypt the transmission data via symmetric encryption. A detailed implementation of this process is shown in Fig. 5 and described as follows:

Step 1: The sender sends the $h(K_c)$ and $E_{Sym_{K_s}}(DATA \| MAC_{K_s}(DATA))$ to receiver.
Step 2: The receiver verifies the $h(K_c)$. If the $h(K_c)$ is not valid, the receiver goes back to sleep mode immediately. If the $h(K_c)$ is valid, the receiver decrypts the data and checks the MAC of data.
Step 3: The receiver sends the data ACK to sender.

Hence, the sender computes $h(K_c)$ from known $K_s$ or $K_c$. To check the received packet valid, the receiver only compares $h(K_c)$. If the $h(K_c)$ is not valid, the receiver goes back to sleep mode immediately and discards all the rest processes. It is infeasible to compute the $h(K_c)$ from $h(K_c)$. The sender must compute $h(K_c)$ from known $K_s$ or $K_c$. The hash chain $h(K_c)$ and $h(h(K_c))$ authenticate sender and receiver mutually.
VI. SECURITY ANALYSIS

A. Mutual Authentication:

The dynamic session key $K_s$ is a hash function and includes 3 items: $K_c$, $R_s$, and $R_r$. In these items, the $R_s$ and $R_r$ are newly selected random numbers by the sender and receiver respectively. These random numbers will be changed every time to ensure the $K_s$ to be created dynamically. The cluster key $K_c$ is shared only by the valid member nodes of a cluster, which indicates that the sender and receiver are valid member nodes of cluster. Thus, the sender and receiver can be authenticated mutually.

B. Secure Token Replay Attack:

In Tier-1, an anti-node may replay the previous eavesdropped random number $R_s$ and secure token $h(K_c \mid R_s)$ as a fake preamble to the receiver. Since the random number $R_s$ and secure token $h(K_c \mid R_s)$ are created dynamically during the transmission session, they are different in every session. The receiver can record and ignore the recent $R_s$ and $h(K_c \mid R_s)$ to resist the repeatedly secure token replay attack. Note that the number of recorded recent $R_s$ and $h(K_c \mid R_s)$ may depend on the memory amount of sensor nodes.

C. Forge Attack:

Without known $K_c$, an anti-node cannot compute the new dynamic session key $K_s$. It is infeasible to compute the $h(K_c)$ from $h(h(K_c))$ nor to compute the $K_s$ from $h(K_c)$.

1) Fake Preamble ACK Attack:

In Tier-1, an anti-node may send a fake preamble ACK to deceive the sender to keep sending data and therefore consuming energy. Since the receiver must compute $h(h(K_c))$ from $K_s$, the valid $h(h(K_c))$ can be used to protect the sender against fake preamble ACK attack from anti-node.

2) "Garbage" Data Attack:

In Tier-2, an anti-node may send "garbage" data to cheat the receiver to go into the step of decrypting data and therefore consuming energy. Since the sender can not compute the $h(K_c)$ from received $h(h(K_c))$, the valid $h(K_c)$ can protect the sender against "garbage" data attack from an anti-node.

VII. ENERGY ANALYSIS

In order to analyze the performance of the TE$_S$S, the operating mode of MCU and radio need to be identified. Here, we consider the energy consumption of sensor node, including the MCU and radio module. Fig. 6 and 7 depict the state diagrams of the TE$_S$S in roles of receiver and sender respectively.

A. Period of sleep

During the sleep mode of the MCU and radio are both in the sleep mode. The energy consumption of a sender or receiver is:

$$E_s = E_R = (P_{MS} \times T_s) + (P_{RI} \times T_S)$$

B. Preamble computing step

In Step 1 of Tier-1, the sender groups the messages and forms the preamble. The sender's MCU is active for processing the security procedure and its radio is in idle mode. The energy consumption of sender is given by:

$$E_s = (P_{AC} \times T_{AC}) + (P_{RI} \times T_{AC})$$

C. Preamble transmitting step

In Step 2 of Tier-1, the sender sends preamble repeatedly before the preamble ACK is received. The sender's radio is in transmit mode and then listens to the receiver for the preamble ACK periodically. The MCU keeps in active mode during this step. Thus, the energy consumption of sender is:

$$E_s = (\text{energy of active mode MCU} + \text{energy to send preamble} + \text{energy to listen ACK}) \times (\text{expected preamble required})$$

$$= (P_{AC} \times (T_p + T_{PAL})) + (P_{R} \times T_p) + (P_{RI} \times T_{PAL})) \times \left( \frac{T_w + T_s}{T_w - T_p} \times (1 - \rho) \right)$$
The factor $\rho$ denotes the probability of receiving a packet in any given interval [6]. In the last preamble period, the receiver is awakened by schedule and receives the preamble. Now, the waked up receiver's MCU is in active mode and the radio is in receive mode to receive the preamble. The energy consumption of receiver yields:

$$E_R = (P_{AC} \times T_p) + (P_{RX} \times T_p)$$

D. Computing and listening steps
In Step 5 of Tier-1 and Step 1 of Tier-2, for processing the security procedure, the sender's MCU keeps active to compute and its radio can go into idle mode. The energy consumption of sender is given by:

$$E_S = (P_{AC} \times T_{AC}) + (P_{RH} \times T_{AC})$$

Now since the receiver is listening to the radio signal, the receiver's radio is in receive mode. The listening duration of receiver should be the same as the computing time of sender. The energy consumption of receiver is given by:

$$E_R = (P_{AC} \times T_L) + (P_{RX} \times T_L) = (P_{AC} \times T_{AC}) + (P_{RX} \times T_{AC})$$

Notice that in both Step 3 of Tier-1 and Tier-2, the roles of sender and receiver are exchanged.

E. Sending and receiving steps
In Step 2 of Tier-2, the sender transmits a signal to the receiver. The sender's radio is in transmit mode whereas the receiver's radio is in receive mode. The MCUs of sender and receiver keep in active mode at the same time. The energy consumptions of a sender and receiver are:

$$E_S = (P_{AC} \times T_{TX}) + (P_{TX} \times T_{TX})$$

$$E_R = (P_{AC} \times T_{RX}) + (P_{RX} \times T_{RX})$$

Similarly, in both Step 4 of Tier-1 and Tier-2, the roles of sender and receiver are exchanged.

VIII. SIMULATION RESULTS
In this section, we present the MATLAB® simulations on the energy consumptions of X-MAC and TE2S. Here we refer the MicaZ mote node datasheet for analyzing energy consumption. The MicaZ has 4 KB of RAM, 128 KB of ROM and equips with ATmega128L MCU and Chipcon CC2420 radio module [19]. The ATmega128L runs at 8MHz clock and the transmitting data rate of CC2420 is 250 kbps. We assume that the power supply of MicaZ is fixed at 3.0 V. Table I shows the states and energy consumption referred to the datasheets of ATmega128L [20] and CC2420 [21].

<table>
<thead>
<tr>
<th>Devices</th>
<th>States</th>
<th>Current (mA)</th>
<th>Energy Consumption (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATmega128L</td>
<td>Active mode</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>ATmega128L</td>
<td>Idle mode</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>CC2420</td>
<td>Transmit mode (Tx)</td>
<td>17.4</td>
<td>52.2</td>
</tr>
<tr>
<td>CC2420</td>
<td>Receive mode (Rx)</td>
<td>19.7</td>
<td>59.1</td>
</tr>
</tbody>
</table>

In this simulation, the MD5 is chosen as the hashing function, and the RC4 is chosen as the encryption/decryption algorithm because they are computational cheap and suitable for sensor network. The security mechanisms overhead of MD5 and RC4 on ATmega128 have been studied in [15]. The ATmega128L shares the same hardware architecture with the ATmega128, except the ATmega128L runs at 8MHz clock, which is a half of the ATmega128. Therefore the execution time of security algorithm of ATmega128L doubles that of ATmega128 [15]. Table II shows the execution time and energy consumption of security algorithm of ATmega128 and ATmega128L.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Plaintext (bytes)</th>
<th>Action</th>
<th>Execution Time (ms)</th>
<th>Energy (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD5</td>
<td>1-26 Digest</td>
<td>1.473</td>
<td>2.946</td>
<td>70.704</td>
</tr>
<tr>
<td>RC4</td>
<td>16 Init.</td>
<td>0.472</td>
<td>0.944</td>
<td>22.656</td>
</tr>
</tbody>
</table>

We assume the length of cluster key, secure token, random number and session key is 16 bytes. The length of ID and timer of node is assumed 2 bytes. The random number selection is simplified as the hashed output of node ID and MCU's timer. Based on these assumptions, the individual energy consumption of security operations is evaluated. The comparison of 16 bytes secure token and hash chains can be completed in tens of cycles by the MicaZ MCU. Compare with thousands cycles overhead of security algorithms [15], we assume the energy of comparison is slight and can be neglected.

<table>
<thead>
<tr>
<th>TABLE III. PROTOCOL PARAMETERS VALUES OF SIMULATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
</tr>
<tr>
<td>Duration of node sleep ($T_d$)</td>
</tr>
<tr>
<td>Duration of node awake ($T_w$)</td>
</tr>
<tr>
<td>Duration of preamble transmitting ($T_P$)</td>
</tr>
<tr>
<td>Duration of preamble ACK transmitting ($T_{Pal}$)</td>
</tr>
<tr>
<td>Duration of Preamble ACK ($T_{Pal}$)</td>
</tr>
<tr>
<td>Maximum duration of preamble transmitting</td>
</tr>
<tr>
<td>Duration of data transmitting ($T_d$)</td>
</tr>
<tr>
<td>Duration of data ACK transmitting ($T_{Pal}$)</td>
</tr>
<tr>
<td>Duration of idle listening before sleep</td>
</tr>
</tbody>
</table>
Table III shows the protocol parameters values used in the simulations. The duty cycle is 4.76%. The transmitted data will be encrypted by RC4 algorithm in the original X-MAC protocol. The packet sending rate is 1 packet every 5 seconds and 3 seconds. The sensor node will send out received data right after the data packet received and completed. Fig. 8 shows the simulation results under the packet sending rate of 1 packet every 5 seconds and 3 seconds respectively. Table IV shows the normalized energy consumptions of original X-MAC and the proposed secure TE$_2$S scheme.

![Energy consumption under the packet sending rate of 1 packet every 5 seconds and 3 seconds](image)

**Figure 8.** Energy consumption under the packet sending rate of 1 packet every 5 seconds and 3 seconds

<table>
<thead>
<tr>
<th>Packet rate</th>
<th>Energy consumption (J/Minute)</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X-MAC</td>
<td>TE$_2$S</td>
</tr>
<tr>
<td>1 packet / 5 seconds</td>
<td>24.78</td>
<td>25.59</td>
</tr>
<tr>
<td>1 packet / 3 seconds</td>
<td>33.67</td>
<td>35.04</td>
</tr>
</tbody>
</table>

Under the packet sending rate of 1 packet every 5 seconds, the proposed secure TE$_2$S scheme increases 3.27% in energy consumption. With the packet sending rate 1 packet every 3 seconds, the proposed secure TE$_2$S scheme increases 4.08% in energy consumption.

**IX. CONCLUSION**

This paper proposes a cross-layer design of energy-efficient secure scheme integrating the MAC protocol. No extra packet is involved in the original MAC protocol design. This scheme can reduce the authenticating process as short as possible to mitigate the effect of the power exhausting attacks. The security analysis shows that this scheme can counter the replay attack and forge attack. The energy analysis identifies the operating mode precisely, including the MCU and radio modules. The simulation result of normalized energy consumption shows that the proposed scheme increases 4.08% in energy consumption, under the packet sending rate of 1 packet every 3 seconds. The energy analysis shows that this scheme is efficient. Further energy consumption of the proposed scheme under various data packet rate and attack scenarios will be investigated in the future. More LPL based WSNs MAC protocols other than X-MAC, such as B-MAC, will be adopted to provide more extensive simulation results to support the efficiency of TE$_2$S scheme. The evaluation will also extend from single node to multiple nodes.

**REFERENCES**


