Two-Tier Energy-Efficient Secure Scheme for Hierarchical 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, mutual authentication, 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].

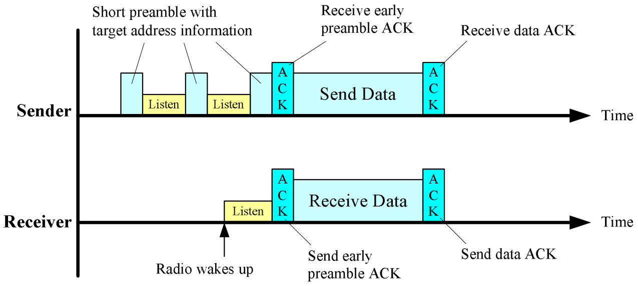

Figure 1. Timeline of X-MAC protocol

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 (TE_2S), 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 suit 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.

- ID_X: X's identity.
- K_s: session key.
- K_c: cluster key.
- R_s, R_r: random number selected by sender and receiver respectively.
- $h(x)$: a one-way hash function which x is the input.
- $E_K(x)$: encrypts x by using symmetric algorithm with key K.
- $D_K(x)$: decrypts x by using symmetric algorithm with key K.
- $MAC_K(x)$: message authentication function with key K, where x is the input message.
- P_{TX}: power of radio in transmit mode.
- P_{RX}: power of radio in receive mode.
- P_{RI}: power of radio in idle mode.
- P_{RS}: power of radio in sleep mode.
- P_{AC}: power of MCU in active mode.
- P_{MS}: power of MCU in sleep mode.
- T_S: duration of node sleep.
- T_W: duration of node awake.
- T_P: duration of preamble.
- T_{PA}: duration of preamble ACK.
- T_{PAL}: duration of preamble ACK listening.
- T_L: duration of listening in steps.
- T_{AC}: duration of MCU active and computing in steps.
- T_{TX}: duration of radio transmitting in steps.
- T_{RX}: duration of radio receiving in steps.
- | : this vertical bar is used to denote concatenation of strings.

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].

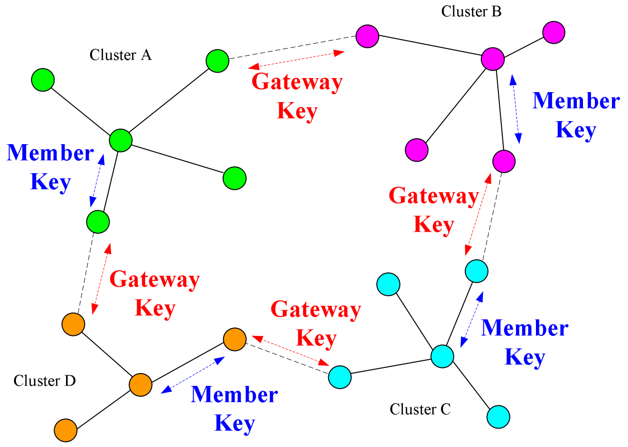

Figure 2. Key distribution for WSNs

In phase I, an authenticated broadcasting mechanism is applied to identify the anti-nodes. In phase II, the adaptive distributed topology control algorithm (ADTCA) [18] performs the clusterhead selection and the gateway selection to form the

Identify applicable sponsor/s here. If no sponsors, delete this text box. (sponsors) 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 TE$_2$S

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.

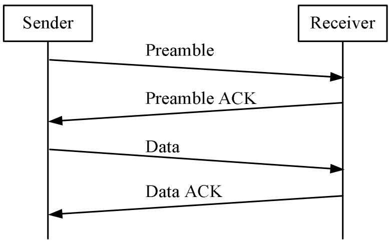

Figure 3. Packet exchange behavior in the X-MAC protocol

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. $Token = h(K_c \mid R_s)$).

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 $K_s = h(K_c \mid R_s \mid R_r)$. The receiver also computes the hash chain $h(K_s)$ and $h(h(K_s))$.

Step 4: The receiver sends the $h(h(K_s))$ and random number R_r as the ACK.

Step 5: The sender computes the session key $K_s = h(K_c \mid R_s \mid R_r)$ and the hash chain $h(K_s)$ and $h(h(K_s))$. The sender then verifies the $h(h(K_s))$. If the $h(h(K_s))$ is not valid, the sender will not send the data.

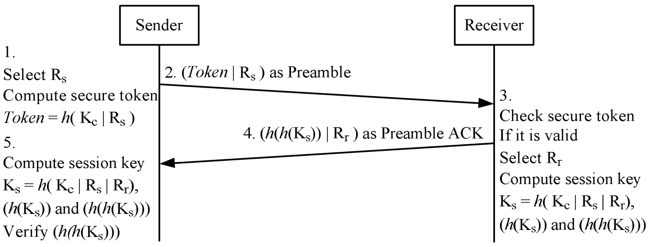
Sender
Receiver
1.
Select R_s
Compute secure token
$Token = h(K_c \mid R_s)$
2. ($Token \mid R_s$) as Preamble
3.
Check secure token
If it is valid
Select R_r
Compute session key
$K_s = h(K_c \mid R_s \mid R_r)$,
$(h(K_s))$ and $(h(h(K_s)))$
4. $(h(h(K_s)) \mid R_r)$ as Preamble ACK
5.
Compute session key
$K_s = h(K_c \mid R_s \mid R_r)$,
$(h(K_s))$ and $(h(h(K_s)))$
Verify $(h(h(K_s)))$

Figure 4. Session Key Agreement

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 $h(h(K_s))$. If the $h(h(K_s))$ is not valid, the sender will not send the data. The hash chain $h(K_s)$ and $h(h(K_s))$ 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 $h(K_s)$ and $E\_Sym_{Ks}(DATA \mid MAC_{Ks}(DATA))$ to receiver.

Step 2: The receiver verifies the $h(K_s)$. If the $h(K_s)$ is not valid, the receiver goes back to sleep mode immediately. If the $h(K_s)$ is valid, the receiver decrypts the data and checks the MAC of data.

Step 3: The receiver sends the data ACK to sender.

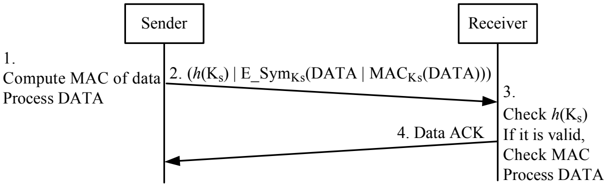

Figure 5. Data Transmission

Hence, the sender computes $h(K_s)$ from known K_s or K_c. To check the received packet valid, the receiver only compares $h(K_s)$. If the $h(K_s)$ is not valid, the receiver goes back to sleep mode immediately and discards all the rest processes. It is infeasible to compute the $h(K_s)$ from $h(h(K_s))$. The sender must compute $h(K_s)$ from known K_s or K_c. The hash chain $h(K_s)$ and $h(h(K_s)$ 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 can not compute the new dynamic session key K_s. It is infeasible to compute the $h(K_s)$ from $h(h(K_s))$ nor to compute the K_s from $h(K_s)$.

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 $h(h(K_s))$ from K_s, the valid $h(h(K_s))$ 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 $h(K_s)$ from received $h(h(K_s))$, the valid $h(K_s)$ can protect the sender against "garbage" data attack from an anti-node.

VII. Energy Analysis

In order to analyze the performance of the TES, 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.

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:

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

$$= \left((P_{AC} \times (T_P + T_{PAL})) + (P_{TX} \times T_P) + (P_{RX} \times T_{PAL})\right) \times \left(\frac{T_W + T_S}{T_W - T_P} \times (1-\rho)\right)$$

The factor ρ denotes the probability of receiving a packet in any given interval [6]. In the last preamble period, the receiver is awaked up 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:

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

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. Numerical Evaluation

In this section, we present the numerical evaluations on the energy consumptions of secure communication between a pair of sender and receiver. 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 I. Energy state of ATmega128L and CC2420

Devices	States	Current	Energy Consumption
ATmega128L	Active mode	5 *m*A	15 *m*W
	Idle mode	2 *m*A	6 *m*W
	Sleep mode	25 μA	0.075 *m*W
CC2420	Transmit mode (Tx)	17.4 *m*A	52.2 *m*W
	Receive mode (Rx)	19.7 *m*A	59.1 *m*W
	Idle mode	20 μA	0.06 *m*W
	Sleep mode	1 μA	0.003 *m*W

In this evaluation, 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 II. Execution time and energy consumption of security algorithm

Algorithm	Plaintext (bytes)	Action	Execution Time (*ms*)		Energy (μJ)
			Atmega128	*Atmega128L*	
MD5	1-26	Digest	1.473	2.946	70.704
	62-80	Digest	2.722	5.444	130.656
RC4	16	Init.	0.472	0.944	22.656
		Enc./Dec	0.086	0.172	4.128

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 III. Protocol parameters values of simulations

Parameters	Time (*ms*)
Duration of node sleep (T_S)	500
Duration of node awake (T_W)	25
Duration of preamble transmitting (T_P)	2
Duration of preamble ACK listening (T_{PAL})	20
Duration of Preamble ACK (T_{PA})	2
Maximum duration of preamble transmitting	500
Duration of data transmitting (T_D)	4
Duration of data ACK transmitting (T_{DA})	2
Duration of idle listening before sleep	20

Table III shows the protocol parameters values used in the evaluation. 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 10 seconds and 5 seconds. Table IV shows the normalized energy consumptions of original X-MAC and the proposed secure TE$_2$S scheme. Fig. 6 and 7 show the simulation results under the packet sending rate of 1 packet every 10 seconds and 5 seconds respectively. The sender consumes more energy than the receiver due to the broadcasting of multiple preambles.

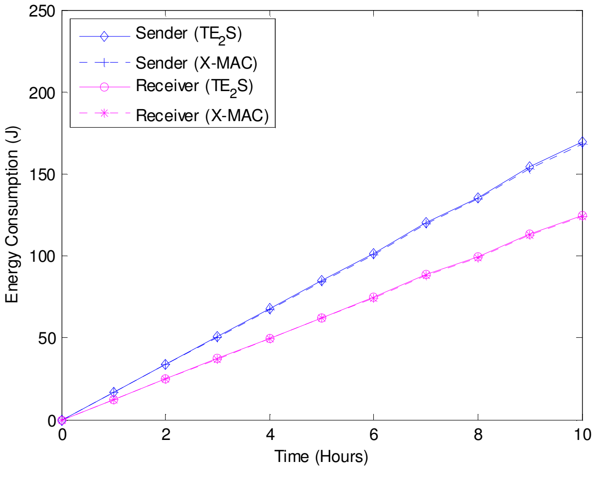

Figure 6. Energy consumption under the packet sending rate of 1 packet every 10 seconds

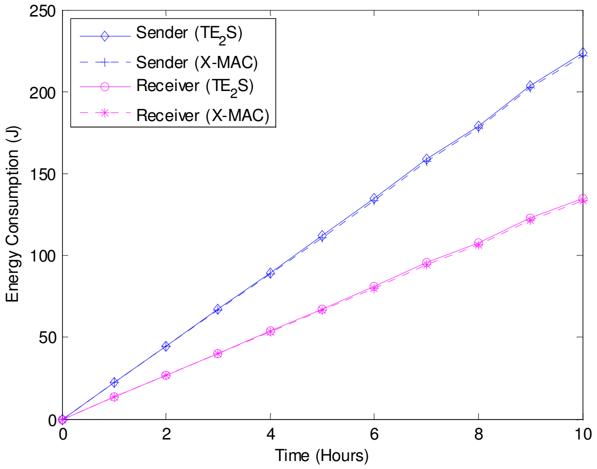

Figure 7. Energy consumption under the packet sending rate of 1 packet every 5 seconds

TABLE IV. NORMALIZED ENERGY CONSUMPTIONS OF SCHEMES

Packet rate	Roles	Energy consumption (J/Hour)		Difference
		X-MAC	*Proposed*	
1 packet / 10 seconds	Sender	16.87	16.98	0.67%
	Receiver	12.42	12.48	0.55%
1 packet / 5 seconds	Sender	22.26	22.44	0.82%
	Receiver	13.37	13.50	1.02%

Under the packet sending rate of 1 packet every 10 seconds, the proposed secure TE_2S scheme increases 0.67% and 0.55% in energy consumption of the sender and receiver, respectively. With the packet sending rate 1 packet every 5 seconds, the proposed secure TE_2S scheme increases 0.82% and 1.02% in energy consumption of the sender and receiver, respectively.

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 numerical evaluation of normalized energy consumption shows that the proposed scheme increases 0.82% and 1.02% in energy consumption of the sender and receiver, respectively, under the packet sending rate of 1 packet every 5 seconds. The energy analysis shows that this scheme is efficient. Thus, the design goals of the proposed energy-efficient secure scheme are achieved.

In the future, we would like to evaluate the energy consumption of the proposed scheme under various data packet rate and attack scenarios. The evaluation will also extend from a pair of nodes to the whole network.

REFERENCES

[1] G. P. Halkes, T. V. Dam, and K. Langendoen, “Comparing energy-saving mac protocols for wireless sensor networks,” ACM Mobile Networks and Applications, vol. 10, no. 5, pp. 783-791, Oct. 2005.

[2] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung, “MAC essentials for wireless sensor networks,” IEEE Communications Surveys & Tutorials, vol.12, no.2, pp. 222-248, Second Quarter 2010.

[3] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient mac protocol for wireless sensor networks,” in Proc. INFOCOM, New York, 2002, pp. 1567- 1576.

[4] T. van Dam and K. Langendoen, “An adaptive energy-efficient mac protocol for wireless sensor networks,” in Proc. ACM SenSys, Los Angeles, 2003, pp. 171-180.

[5] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in Proc. ACM SenSys, Baltimore, 2004, pp. 95-107.

[6] M. Buettner, G. V. Yee, E. Anderson and R. Han, “X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks,” in Proc. ACM SenSys, Boulder, 2006, pp. 307-320.

[7] M. Brownfield, Y. Gupta, and N. Davis, “Wireless sensor network denial of sleep attack,” in Proc. 6th Annu. IEEE SMCIAW, 2005, pp. 356-364.

[8] D. Raymond, R. Marchany, M. Brownfield and S. Midkiff, “Effects of denial of sleep attacks on wireless sensor network MAC protocols,” IEEE Transactions on Vehicular Technology, vol. 58, no. 1, Jan. 2009.

[9] R. Falk, and H.J. Hof, “Fighting insomnia: a secure wake-up scheme for wireless sensor networks,” in Proc. SECURWARE, Athens, 2009, pp. 191-196.

[10] Y. C. Ouyang, R. L. Chang, and J. H. Chiu, “A new security key exchange channel for 802.11 WLANs,” in Proc. IEEE ICCST, Taipei, 2003, pp. 216-221.

[11] Y. C. Ouyang, C. B. Jang, and H. T. Chen, “A secure authentication policy for UMTS and WLAN interworking,” in Proc. IEEE ICC, Glasgow, 2007, pp. 1552-1557.

[12] Y. C. Ouyang, C. T. Hsueh, and H. W. Chen, “Secure authentication policy with evidential signature scheme for WLAN,” Security and Communication Networks, vol. 2, no. 3, May/June 2009, pp. 259-270.

[13] N. Haller, and C. Metz, “A one-time password system,” IETF RFC 2289, Feb. 1998.

[14] D. W. Carman, P. S. Kruus, and B. J. Matt, “Constraints and approaches for distributed sensor network security,” NAI Labs, The Security Research Division, Glenwood, Tech. Rep. #00-010, Sept. 2000.

[15] P.Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu, “Analyzing and modeling encryption overhead for sensor network nodes,” in Proc. ACM WSNA, San Diego, 2003, pp. 151-159.

[16] W. Liu, R. Luo, and H. Yang, “Cryptography overhead evaluation and analysis for wireless sensor networks,” in Proc. WRI CMC, Kunming, 2009, pp.496-501.

[17] C. T. Hsueh, Y. W. Li, C. Y. Wen, and Y. C. Ouyang, “Secure adaptive topology control for wireless ad-hoc sensor networks,” Sensors, vol. 10, no. 2, 2010, pp. 1251-1278.

[18] K. T. Chu, C. Y. Wen, Y. C. Ouyang, and W. A. Sethares, “Adaptive distributed topology control for wireless ad-hoc sensor networks,” in Proc. SensorComm, Valencia, 2007, pp. 378-386.

[19] Crossbow MicaZ datasheet, Crossbow Technology Inc., 2006.

[20] Atmel ATmega 128L datasheet, Atmel Corporation, 2009.

[21] Chipcon CC2420 datasheet, Texas Instruments, 2007.