A Lightweight Security and Energy-Efficient Clustering Protocol for Wireless Sensor Networks

Guangsong Yang[1,2] and Xin-Wen Wu[2(✉)]

[1] Jimei University, Xiamen 361021, FJ, China
[2] Griffith University, Gold Coast 4215, QLD, Australia
x.wu@griffith.edu.au

Abstract. Most applications based on wireless sensor networks (WSN) have devices with constraints of limited energy and computational/storage capabilities. The traditional security mechanisms are not desirable to these applications. A lightweight security and energy-efficient clustering protocol was proposed in this paper to solve the security problem in the clustering-based sensor networks. Firstly, a lightweight security algorithm is proposed to meet the security requirements, which reduces the communication overload by using the transmission key index. Secondly, in the process of clustering, the base station (BS) and cluster head (CH) use lightweight authentication procedure to verify the identities hierarchically, to reduce the risk of attacks from malicious nodes posing as BS or CH. Thirdly, the proposed protocol is analyzed in the aspects of security and energy consumption. Simulation results show that the proposed protocol not only enhances the network security but also improves the energy efficiency.

Keywords: Lightweight security · Energy-efficient protocol · Clustering WSN

1 Introduction

Wireless sensor networks based on clustering methods have been proved to improve system throughput, reduce system delay and save energy. Some clustering protocols, such as Low-Energy Adaptive Clustering Hierarchy (LEACH) [1], solve the problem of energy efficiency by selecting cluster heads periodically. However, the dynamic nature of the topology also brings challenges to the existing security schemes.

Like most routing protocols in WSN, LEACH is vulnerable to a variety of security attacks [2], including interference, deception, replay, and so on. However, because it is a cluster-based protocol, it basically relies on CH for data aggregation and routing. If an intruder pretends to be a CH, intrusion and selective forwarding can be carried out to destroy the network. Moreover, the intruder may inject forged sensor data into the networking in some way.

Many of the security schemes used in the classic computer networks are not suitable for WSN. For example, the scheme based on public key distribution is easy to

J. Zheng et al. (Eds.): ADHOCNETS 2018, LNICST 258, pp. 237–246, 2019.
https://doi.org/10.1007/978-3-030-05888-3_22

be cracked by malicious nodes because of the requirement of the global key. The applications these schemes pose a significant security vulnerability.

There were lots of works about the security based on LEACH [2]. LEAP [3] is a local key allocation scheme among neighbor nodes, which is very effective for static networks. But in LEACH, each round may require a new Key distribution, which is inefficient and infeasible.

S-LEACH is the first modified version of LEACH with cryptographic protection against outsider attacks. In S-LEACH [4], each node has two symmetric keys: a pair of keys shared with the BS, and the last key of the key chain held by BS for authentication broadcast. BS authenticates the broadcast of CH through two simple steps. Each CH broadcast advertising message named *adv*, which include ID of CH and MAC (generate by the shared key between BS and CH), BS compiling a legitimate list of CH from these *adv*, and broadcast it to whole network by using μTESLA scheme. Ordinary member nodes can know which *adv* messages they receive from the legitimate nodes, and then select correct CH.

Based on the S-LEACH, other two protocols have been proposed. One is Sec-LEACH [5], another is MS-LEACH [6]. S-LEACH and SecLEACH were proposed by the same authors. S-LEACH is improved by SecLEACH which is based on a random key distribution scheme. In Sec-LEACH, the communication between nodes is protected by a key pre-allocation scheme. The main idea is to generate a large number of keys and their ID when deploying the network, and then randomly assign a group of keys to each node. Each node also is assigned a pair of keys shared with the BS, which are used in nodes and BS. It used random-key pre-distribution and μTESLA for secure hierarchical WSN with dynamic cluster formation. Sec-LEACH applied random key distribution to LEACH, and introduced symmetric key and one way hash chain to provide confidentiality and freshness. Sec-LEACH provides authenticity, integrity, confidentiality and freshness to communications.

MS-LEACH [6] was proposed to enhance the security of S-LEACH by providing data confidentiality to CH authentication using pairwise keys shared between CHs and their cluster members. It does not provide authentication for join request message. There is no key update provisioning for key. It requires multiple unicast communications. This way the energy of a CH can be depleted.

In this paper we proposed a lightweight security scheme for LEACH (which we call LS-LEACH) based on our previous works [9, 10]. The scheme significantly reduces the security overload and provides a higher level of security for distributed and dynamic sensor networks.

The paper is organized as follows, in Sect. 2 the lightweight security protocol for WSN was described. Section 3 proposed a LS-LEACH security protocol based on LEACH. Section 4 presents LS-LEACH performance evaluation. Finally, the concluding remarks and future work are given in Sect. 5.

2 The Lightweight Security Method for WSN

2.1 The Light Weight Security Method

The lightweight security protocol we proposed in [9] based on the work [10] include the process of lightweight encryption, key management and identity authentication, the security is ensured by the procedure of one-key-for-one-file encryption and the security of the key management. Encryption and decryption are executed using a probabilistic encryption procedure or using hashed key.

Firstly, we should prepare a large key store for legitimate users in advance. The key store seed can be stored in the device's hardware security module [8]. When an attacker physically disrupts the device and trying to extract it without successfully authenticating, it will be deleted by the device automatically.

The Structure of proposed lightweight security algorithm is show in Fig. 1. Let S be the sending party and R be the receiving party. Each key $k(\omega)$ is efficiently generated through the key storage seed K and index ω shared by the users and devices (The key is uniquely determined by ω and the seed K in the storage pool). There is no need to transfer keys or maintain them between devices. From the information theory point of view, the key management program is safe, which means that no information about the key is disclosed when the key index is transmitted. Key generation, allocation and usage are specified as follows:

(1) To encrypt x, the sender first needs to pick a random key index ω and seed K in the key store, then generate an encryption key $k(\omega)$ using an efficient algorithm. $k = (k_1, k_2, k_3, \ldots k_n)$ is a key with n bits, $x = (x_1, x_2, x_3, \ldots x_n)$ is information with n bits, the method of Encryption and Decryption are,

$$\begin{aligned} &\text{Encryption} : E_k(x) = (f_k(r), x \oplus h(r)) \\ &\text{Decryption} : D_k(y, z) = h(f_k^{-1}(y)) \oplus z \end{aligned} \tag{1}$$

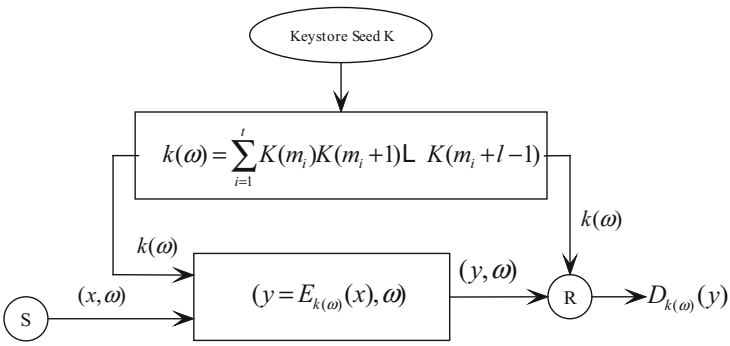

Fig. 1. Structure of proposed lightweight security algorithm

Where, $f_k(x) = (k_1 \oplus x_1, k_2 \oplus x_2, k_3 \oplus x_3, \ldots k_n \oplus x_n)$, $y = E_{k(\omega)}(x), r$ is a random string (independent from the key and plaintext) and h is a random oracle implementation by using a hash function). The seed in key store $K = K(0)K(1)\cdots K(L-1)$ is an L-bit random string, which is used to generate cryptographic keys of length $l(l \ll L)$. Assume $\Omega = (m_1, m_2, m_3, \ldots m_t)$ is a set of cardinality Λ, where t is a positive integer, and $0 \le (m_1, m_2, m_3, \ldots m_t) \le L$. Let the elements of Ω act as key indices. For any of $0 \le \omega = (m_1, m_2, m_3, \ldots m_t)$ in Ω, the Key with length l can be expressed as $k(\omega)$ or $k(m_1, m_2, m_3, \ldots m_t)$, show as

$$k(\omega) = k(m_1, m_2, m_3, \ldots m_t) = \sum_{i=1}^{t} K(m_i)K(m_i+1)\cdots K(m_i+l-1) \tag{2}$$

Where, $\sum_{i=1}^{t} K(m_i)K(m_i+1)\cdots K(m_i+l-1)$, is bit by bit binary addition.

Here, the sum of the integer m, and j, are related to the module L, so the key store $\Psi = \{k(m_1, m_2, \ldots m_t) : 0 \le (m_1, m_2, m_3, \ldots m_t) \le L-1\}$, the number of available keys here is $\Lambda = \begin{pmatrix} L \\ t \end{pmatrix}$

(2) The key index ω with ciphertext (can be placed on the head of the encrypted message packet), are sent to the receiver and encrypted, then $k(\omega)$ are deleted.
(3) The receiver uses ω to regenerate key $k(\omega)$, using the same key index ω and the same key generation process, then used for decryption or authentication verification.

2.2 Identity Authentication Process

When a new legitimate device joins a local network system for the first time, the system administrator configures it through some security method (such as a manual method), so that the device shares it's unique and secret with the HUB or other devices. With this configuration procedure, the new device is also know the identity of other devices. These identities may then be maintained by a hardware security module.

(1) *Identity Authentication*
For any two device D1 and D2,with the unique and secret identity ID_1 and ID_2 respectively. They authenticate each other as follow

Step1. D1 sent the content to D2 as

$$(\omega_1; F_{k(\omega_1)}(ID_1||TS)) \tag{3}$$

Where, ω_1 is a randomly selected key index, TS is a timestamp (used to prevent replay), $F(\cdot)$ is a valid cypher.

Step2. Using the key index ω_1, D2 generates the key $k(\omega_1)$ and decrypts ID_1

$$ID_1||TS = F^{-1}_{k(\omega_1)}(F_{k(\omega_1)}(ID_1||TS)) \tag{4}$$

D2 verifies D1 (obtained from BS or CH) by comparing the decrypted content with the identity of D1;
Step3. Randomly select the key index ω_2 and generate the key $k(\omega_2)$, D2 will send the following content to D1

$$(\omega_2; F_{k(\omega_2)}(ID_2 \oplus ID_1||TS)) \tag{5}$$

Step4. Using the key index ω_2, D1 generates the key $k(\omega_2)$ and decrypts the ID of D2

$$ID_2||TS = ID_1 \oplus F^{-1}_{k(\omega_2)}(F_{k(\omega_2)}(ID_2 \oplus ID_1||TS)) \tag{6}$$

D1 thus obtains the identity of D2.

3 Security Clustering Protocol with Lightweight Method

In this section, a clustering algorithm named LS (Lightweight Security)-LEACH is proposed, which is based on the classical LEACH algorithm combined with our lightweight security method in the process of authentication and clustering.

LEACH protocol employs randomized rotation of the cluster-heads to evenly distribute the energy load among the sensor nodes in the network. The operation of LS-LEACH is similar to LEACH which divided into rounds show as Fig. 2. Each round consists of two phases: a set-up phase and a steady-state phase. During the set-up phase cluster-heads are determined and the clusters are organized. During the steady-state phase data transference to the base station occurs.

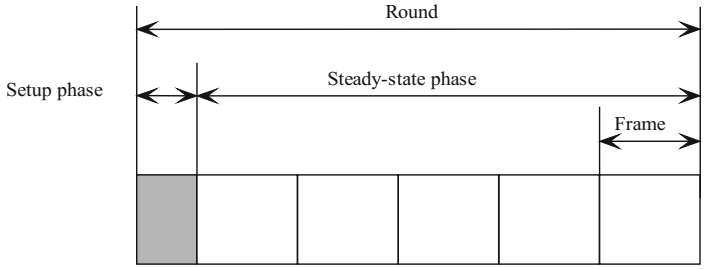

Fig. 2. Timeline of LEACH

The more detailed working process is shown in Table 1.

Table 1. LS-LEACH Protocol.

The algorithm of LS-LEACH Protocol.
Setup phase
Step1. BS $\Rightarrow \Lambda$: $adv_{BS}(ID_{BS}, \omega_{BS})$
CH $\Rightarrow \Lambda$: $req_{BS}(ID_{CH}, \omega_{CH}, ID_{BS}, SL)$
BS: Authenticated CH, update CH_List and broadcast to the whole network.
A_i : Choose CH according to adv_{CH} and adv_{BS}.
Step2. $A_i \rightarrow$CH :$join\_req(ID_{A_i}, ID_{CH}, \omega_{A_i})$
Step3. CH $\Rightarrow \Lambda$: ID_{CH}, $sched(..., \langle ID_{A_i}, t_{A_i} \rangle, ...)$,
Steady-state phase
Step4. $A_i \rightarrow$CH : $(ID_{A_i}, ID_{CH}, d_{A_i}, \omega_{A_i})$
Step5. CH$\rightarrow$ BS : $ID_{CH}, ID_{BS}, G(..., d_{A_i}, ...), \omega_{CH}$

Symbols defined as below,

A_i ,CH, BS: An ordinary node, a cluster head, and the base station, respectively

Λ : The set of all nodes in the network

$\Rightarrow$, $\rightarrow$: Broadcast and unicast, transmissions respectively

ID_X : Node X's id

d_X : Sensing report from node X

ID_X , t_{Xi} : Node X's id and its time slot t_X in its cluster's transmission schedule

adv, join_req, sched : String identifiers for message types

G : Data aggregation function

j : Reporting cycle within the current round

SL : Security level, it depend on the length of ω

1. **Setup phase**

Step 1:

(1) At the beginning of each round, the BS broadcasts $adv_{BS}(ID_{BS}, \omega_{BS})$to the whole network. After each sensing node obtains ω_{BS}, it generates $k(\omega_{BS})$ and decrypts ID_{BS} according to Eq. (2), them compared with ID_{BS} in its own memory at initialization phase, to determine whether it is a real BS.

(2) The self-recommended cluster head sends network access information $req_{BS}(ID_{CH}, \omega_{CH}, ID_{BS}, SL)$ to the BS, and it also can be heard by its neighbor nodes.

(3) BS generate $k(\omega_{CH})$ according to Eq. (2), decrypt out ID_{CH}, and compare with legitimate users in its own database. If an ID_{CH} is satisfied, it means that it is a valid node, then it is listed in the legitimate CH_List of this round, and broadcast $adv_{BS}(ID_{BS}, \omega_{BS}, CH\_List)$ to the whole network.

Step 2: The normal node A_i chooses the closest CH within its coverage based on the RSSI, generates $k(\omega_{CH})$ by ω_{CH}, decrypts ID_{CH} according to Eq. (3). Similarly, it generates $k(\omega_{BS})$ by ω_{BS}, decrypts CH_List. By comparing decrypted ID_{CH} and CH_List, it determine whether the cluster head is a legitimate CH. Then select ω_{A_i} from the k-store, and send the *join_req* request to join the cluster. If a node did not take part in any cluster, he will communicate to BS directly.

Step 3: CH validates the A_i. Then send the TDMA scheduling information to its member nodes.

2. **Steady-phase**

Step 4: A_i send the encrypted monitoring data d_{A_i} to CH by using ω_{A_i}

Step 5: CH aggregate the information from A_i together, and send these data $G(\ldots, d_{A_i}, \ldots)$ and ω_{CH} to the BS. To ensure freshness, ω_{CH} should be updated at each round.

4 Simulation and Evaluation

The security of proposed protocol has been verified in [9, 10]. In this section, we evaluated the energy effectiveness of our scheme through simulation experiments. In the simulation, 100 sensor nodes are randomly distributed in the square region of size 200 m * 200 m and the BS is in the center of this region. The parameters used in the simulation are summarized in Table 2.

Table 2. Simulation parameters.

Parameter	Meaning	Value
n	Size of data packet	4000bit
a	Size of control packet	100bit
E_{DA}	Aggregation energy consumption	5nJ/bit/signal
E_{sedule}	Energy consumption of Schedule	5nJ/bit/signal
E_{init}	Initial energy	10 J
P_r	Receive power	1 mJ /bit
B_t	Threshold of battery	1 J

SecLEACH messages to be 36 bytes long (the default TinyOS message sizes) and LEACH messages to be 30 bytes long. The difference is meant to account for the size

difference between the MAC (8bytes [16]) and CRC (2 bytes [12]) – the former present in SecLEACH, but absent in LEACH; and the latter present in LEACH, but absent in SecLEACH.

In our scheme, the additional energy consumption mainly from setup phase. We set the length of $l = k(\omega)$ = 128(256) bits = 16(32) bytes. According the analysis in Sect. 2.1, the length of ω can be selected with different security level.

The average of ten times simulation results is show as Figs. 3 and 4.

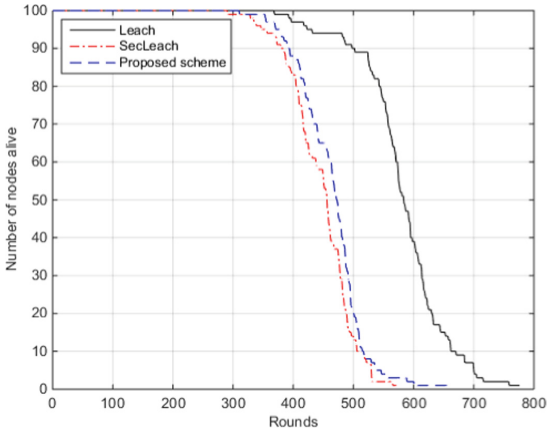

Fig. 3. Dead nodes comparison of different protocol.

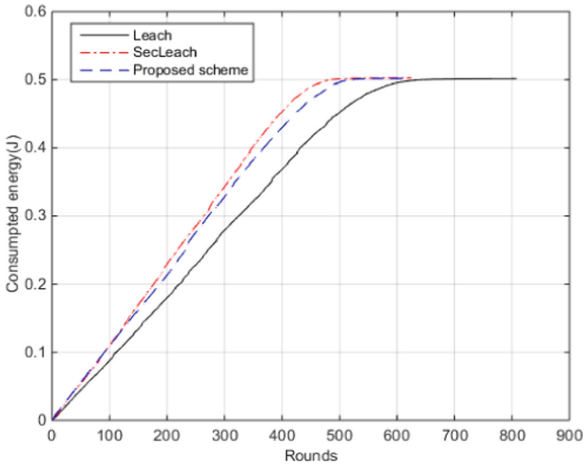

Fig. 4. Energy consumption comparison of different protocol.

Figure 3 shows number of alive nodes using different protocols as rounds varies, We can easily observe that the lifetime of LEACH is longer than that of LS-LEACH and SecLEACH. It is due to some load must be added to ensure safety, nodes cost more energy in certain round leads their death in advance. But LS-LEACH performance is better compared to SecLEACH, because the overload is less than that in SecLEACH.

Figure 4 clearly depicts that LS-LEACH outperforms SecLEACH in terms of energy consumption.

5 Conclusion

In this paper, we proposed a lightweight security leach protocol to enhanced the security and minimize the energy consumption of sensor nodes.

Our contribution in this paper are show as bellow,

(1) We propose a Lightweight Security LEACH (LS-LEACH) protocol, to reduce the overload (both encryption, Decrypt, identity authentication and transmission). Due to the lightweight encryption method, only a few indices need to be sent during node authentication process, so the load is reduced and energy efficiency is improved. Through the node authentication, the multiple ID Sybil attacks and wormhole attacks by malicious node are avoided.
(2) The introduced security mechanism is suitable for distributed scenarios, which enhance the security between nodes authentication, meanwhile avoids energy consumption of distant nodes due to direct transmission.
(3) The level can be controlled by adjust the length of index and key, neighbor nodes can communicate between each other based on the security level.
(4) Due to the network password are update in every round, the freshness is Ensured. Future research work includes how to further to improve the energy efficiency of this protocol by using multi-hop and other method.

Acknowledgement. This research is supported by Science Foundation of Fujian Province (No. 2015J01267), Training Program of FuJian Excellent Talents in University.

References

1. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference System Sciences, pp. 10–15 (2000)
2. Rahayu, T.M., Lee, S.G., Lee, H.J.: Survey on LEACH-based security protocols. In: Advanced Communication Technology (ICACT), pp. 304–309 (2014)
3. Zhu, S., Setia, S., Jajodia, S.: LEAP+: efficient security mechanisms for large-scale distributed sensor networks. ACM Trans. Sens. Netw. (TOSN) **2**(4), 500–528 (2006)
4. Ferreira, A.C., Vilaça, M.A., Oliveira, L.B., Habib, E., Wong, H.C., Loureiro, A.A.: On the security of cluster-based communication protocols for wireless sensor networks. In: Lorenz, P., Dini, P. (eds.) ICN 2005. LNCS, vol. 3420, pp. 449–458. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31956-6_53

5. Oliveira, L.B., Wong, H.C., Bern, M., Dahab, R., Loureiro, A.A.F.: SecLEACH-A random key distribution solution for securing clustered sensor networks. In: Fifth IEEE International Symposium on Network Computing and Applications, NCA 2006, pp. 145–154 (2006)
6. El_Saadawy, M., Shaaban, E.: Enhancing S-LEACH security for wireless sen-sor networks. In: 2012 IEEE International Conference on Electro/Information Technology (EIT), pp. 1–6 (2012)
7. Jolfaei, A., Wu, X.W., Muthukkumarasamy, V.: A secure lightweight texture encryption scheme. In: Huang, F., Sugimoto, A. (eds.) PSIVT 2015. LNCS, vol. 9555, pp. 344–356. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30285-0_28
8. Paverd, A.J., Martin, A.P.: Hardware security for device authentication in the smart grid. In: Cuellar, J. (ed.) SmartGridSec 2012. LNCS, vol. 7823, pp. 72–84. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38030-3_5
9. Wu, X.-W., Yang, E.-H., Wang, J.: Lightweight security protocols for the internet of things. In: Proceedings of the 28th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC 2017), Montreal, Canada, 8–13 October 2017
10. Yang, E.H., Wu, X.-W.: Information-theoretically secure key generation and management. In: Proceedings of 2017 IEEE International Symposium on Information Theory, pp. 1529–1533 (2017)
11. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Comput. Netw. **52**(12), 2292–2330 (2008)