

A New Scheme of Key Distribution using Implicit Security in Wireless Sensor Networks

Chien-Wen Chiang, Chih-Chung Lin, Ray-I Chang

Department of Engineering Science, National Taiwan University, Taipei, 10617, Taiwan
{R97525036, R97525035, rayichang}@ntu.edu.tw

Abstract— Wireless Sensor Network (WSN) is a noticeable subject currently. There are many WSN applications including military control and environmental monitoring. In these applications, the security of data transmission is an important issue. Because sensor nodes are resource limited and they do not have fixed structure model, there needs an efficiently key distribution and management mechanisms to achieve secure communication. This paper proposes a scheme which uses an implicit security technique to distribute the encrypted key. Because most of operations are computed by the base station, this scheme can extend survival time of the network and it is very appropriate for WSN.

Keywords— Implicit Security, Key Distribution, Wireless Sensor Networks, Key Management, Security Requirements

I. INTRODUCTION

A Wireless Sensor Network (WSN) is constructed by a group of sensor nodes with characteristics of tiny shape, low-energy consumption and cheap cost. In general, sensor nodes are distributed in a certain area. Most of WSN has one or more base stations which may be a laptop or desktop computer and they have higher computing capability and bandwidth than other sensor nodes. After sensor nodes collect information from the environment, they deliver these data to the base station by multi-hop. Then the data will be operated or sent to client for analyzing. Because WSN does not need to pre-build infrastructures, it has wide range of applications involving military control, environmental monitoring, healthcare, home automation and other commercial applications [1].

When the routing protocol is initiated, the sensor nodes will build their routing paths. Data are delivered between nodes by following these routing paths until they arrive to the base station. Because data are transmitted through wireless communication, they might be eavesdropped or modified by malicious devices if not be protected. Unpredictable damages may be caused when the secret information is leaked [2]. For example, if the system is operated in the military tracking and the military information of transmission is leaked, it is dangerous for determine the situation of the battlefield. In the other hand, we consider the system is operated in the health care. If the personal private information (personal medical information) is intercepted, it will cause many social problems. For these reasons, ensure that the data can arrive to destination correctly and data validity in destination becomes an important issue [3][4].

Modern networks use encryption techniques (especially

for asymmetric encryption [5]) to protect the privacy and integrity of data. However, current asymmetric encryption methods usually require a large quantity of computations. Due to inherent characteristics of WSN, sensor nodes have a lot of limitations in storage, computing capability and bandwidth. These methods are inappropriate for WSN. Although symmetric encryption methods [6][7] are more appropriate for WSN, they need a key management protocol to distribute keys into nodes for providing the communication nodes a same session key.

This paper proposes a scheme which used implicit security to partition the key to many parts. And then the base station assigns these key partitions to every nodes of the network. When a node wants to return the data to the base station, node will recreate these key partitions to an encrypted key and uses this key to encrypt the data. The scheme can reduce the overhead of computing and storage of nodes, hence it can save the time and the energy of encrypting data. The scheme is appropriate for WSN due to the scheme can extend survival time of the network.

II. RELATED WORKS

Because of the data of transmission between nodes may be eavesdropped, the data must be encrypted to ensure the security. Therefore, the scheme of key distribution has become the most important key. In general, there are three key distribution schemes: trusted-server scheme, public-key scheme and key pre-distribution scheme [8].

The key distribution and management of WSN may be subject to the following constraints [1].

Communication Security Constraints: Because of the energy, computing capability, bandwidth and storage constraints of sensor nodes, the public-key scheme that needs a large quantity of computations like RSA [9] is inappropriate for WSN [10][11]. In addition, a large amount of computation of public-key scheme will make a large number of energy consumption when the network is subjected to denial-of-service (DoS). It will lead the entire network to crash. Therefore, in the design of key distribution is still symmetric encryption, low computation of the authentication function and the hash function as the main consideration.

Key Management Constraints: The density of a large-scale WSN may be more than ten times or hundred times than a traditional wired network. Moreover, as we do not know the topology of WSN before nodes are deployed, it is inappropriate

for key distribution by trusted devices (or a Key Distribution Center, KDC).

As the two restrictions above, and characteristics of nodes that the communication range is limited, low computing capability. It is difficult to operate key distribution and key management mechanisms after deployment of nodes. Therefore, Eschenauer and Gligor believed that the key pre-distribution scheme would be the most suitable way for a large-scale WSN [12].

In past years, different key distribution and key management mechanisms were proposed for WSN [3][13] (such as the single network-wide key scheme, the fully pair-wise shared key scheme [14][15], and the random key pre-distribution scheme [12]).

A. Single Network-Wide Key Scheme

Single network-wide key scheme is the simplest way to distribute keys. It pre-loads a same key to each node before network deployment. Therefore, it uses this key to communicate with other nodes and add a message authentication code (MAC) to confirm the integrity of the data. The disadvantage of the scheme is that if there one node is compromised, and then all the connections links are insecure. In addition, it cannot provide the authentication of individual nodes due to all the nodes use the same key.

B. Fully Pairwise-Shared Key Scheme [16]

Fully pairwise-shared key scheme is a scheme which shares a session key between any two nodes. Suppose there are n nodes of the network, then the total number of key in the system are $\binom{n}{2}$ and each node have to store $n-1$ keys. It is safer than the single network-wide key scheme and when a node is compromised, the network will not be broken. But storing a large number of keys does not meet the actual demand due to the sensor node will only communicate with pre-specified neighbors.

C. Random Key Pre-distribution Scheme

The random key pre-distribution scheme proposed by Eschenauer and Gligor [12] is widely researched in recent years as it has higher security. It establish a key pool (suppose the key pool has p keys), and it will select m keys to establish a key ring. When the wireless sensor deployment is complete, each sensor node will has its own key ring identification (Key Ring ID) information and broadcast out it. If two nodes within the range of communication, they will establish a secure communication link and share a same secret key. However, it may cause two communication links to have the same session key and makes some securities and certifications problems. In addition, since each node had been distributes number of the keys, if a node has been compromised, the attacker will gain more secret data.

Parakh and Kak [17] proposed a scheme which can divide the data into two or more parts by determining all roots of a polynomial in a finite field Z_p (p is a large prime),

and these roots are data partitions. These partitions are stored on randomly-chosen servers on the network, and only the owner of data known where these partitions are stored. It needs to access all the partitions to reconstruct data. The illustration is shown in Figure 1. The user divides data into k partitions. It can also partition the key into servers on the network. User has to reconstruct key partitions to be a decrypt key firstly, and use the key to decrypt the data from another server. It is shown in Figure 2.

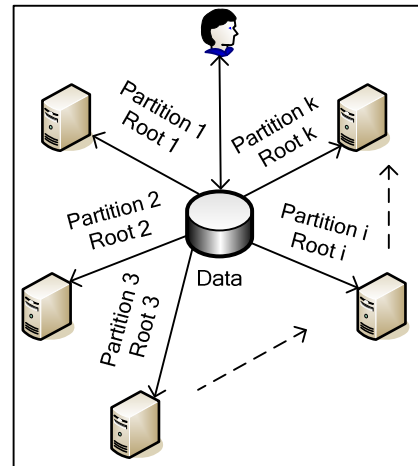


Figure 1. The illustration of data partitions [11]

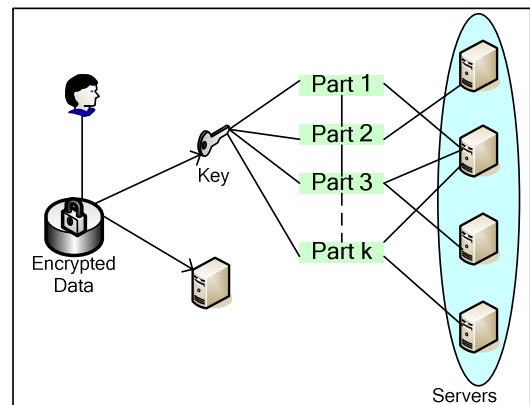


Figure 2. The illustration of key partitions [11]

III. THE NEW KEY DISTRIBUTION SCHEME

This paper proposes a scheme of key distribution which partitions the key into k parts and distributes these partitions over the nodes on WSN. When data want to return to the base station, it will reconstruct key partitions of the path to be a new encrypted key. The node uses this key to encrypt data by symmetric encryption. As our scheme takes the most part of computation on the base station, sensor nodes take lower computation to reducing the encrypting time and the number of keys. It increases the survival time and decreases the response time of the network.

Parakh and Kak use the roots of polynomial shown in (1) and (2) to partition the data.

$$x^k + a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \dots + a_1x + d = 0 \quad (1)$$

$$\prod_{i=1}^k r_i \equiv d \pmod{p} \quad (2)$$

In this paper, we propose a new way to partition the key for key distribution. First, base station will create a key randomly, and the key is partitioned to N (number of nodes on WSN) parts. The roots of (3) are our key partitions. When key partitions are built, the base station will distribute these partitions into nodes randomly (every node just has one key partition), and record where these partitions are stored.

$$(x-r_1)(x-r_2)(x-r_3)\dots(x-r_N) \equiv 0 \pmod{p} \quad (3)$$

When user executes the query, the base station will broadcast this query packet to each node. This query packet record the path information (through which nodes). The illustration is shown in Figure 3 where *Query* is user broadcast query packet, K_p are key partitions. As there are six nodes, we have six key partitions.

We assume the node E want to return data to base station. It returns a acknowledge packet (Ack) and requests key partitions of the path between base station and itself as shown in Figure 4. The node E uses the key partitions $k_{temp} \equiv k_{p1} \cdot k_{p2} \cdot k_{p4} \pmod{p}$ to build the session key $k_{session} \equiv k_{temp} \cdot k_{p6} \pmod{p}$ in Figure 5. The node E hashes the data for verify the integrity of the data. Finally, node E encrypt data and the result after hashed, and sent encrypted data to base station as shown in Figure 6.

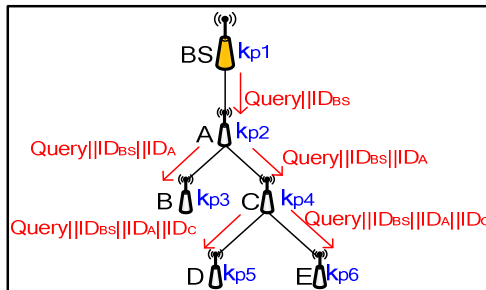


Figure 3. Distribution of key partitions and delivery of query packets

Because of the base station recorded the key partition of every nodes, it can calculate the session key of communicate node quickly. When the base station decrypts the data, it also verifies the integrity of the data by executes the hash function to the data. Furthermore, the base station can request key partitions of every node frequency. It can detect the current status of the network by compare the reconstruct of key form nodes and its records.

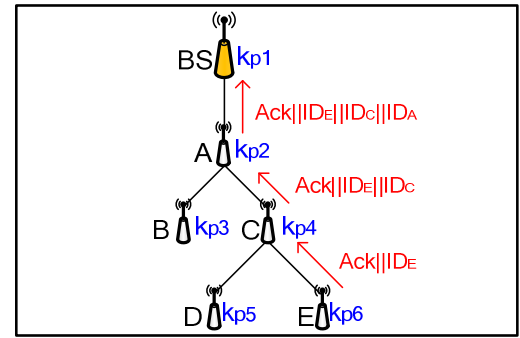


Figure 4. Sensor nodes request key partitions of the path from the base station

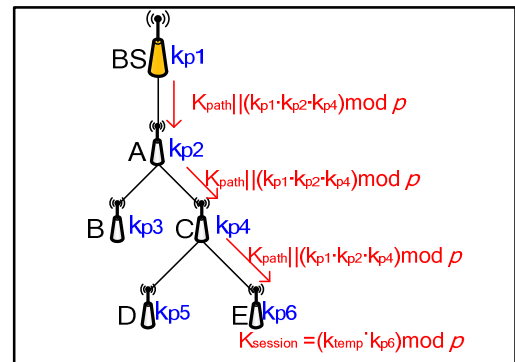


Figure 5. The base station sends the reconstruct of key partitions of the path to the node E

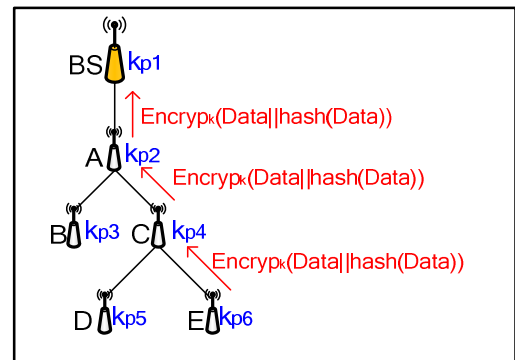


Figure 6. The node E sends the encrypting data to base station

For example, assume there are six nodes on network. The base station creates a key k randomly, let it be 126. We partition k to six parts, and the prime p is 89. We can get the polynomial as follows.

$$x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + 126 \equiv 0 \pmod{89}$$

The roots by the polynomial are as follows.

$$(x-r_1)(x-r_2)(x-r_3)(x-r_4)(x-r_5)(x-r_6) \equiv 0 \pmod{89}$$

The base station selects five random numbers, $r_1 = 12$, $r_2 = 9$, $r_3 = 30$, $r_4 = 24$, and $r_5 = 18$.

These roots are smaller than p .

$$r_6 \equiv k \cdot (r_1 \cdot r_2 \cdot r_3 \cdot r_4 \cdot r_5)^{-1} \mod 89 \equiv 10$$

polynomial become

$$(x-12)(x-9)(x-30)(x-24)(x-18)(x-10) \equiv 0 \mod 89$$

We have

$$a_5 = -14, a_4 = 76, a_3 = -18, a_2 = 39, \text{ and } a_1 = -58,$$

where $r_1 \sim r_6$ are our key partitions.

Table 1. The description of notation

Notation	Description
k_p	Key partitions $k_{p1} \sim k_{pn}$
k_{part}	The part of the path key which is computed by the base station. It still not combines with the key partition with the node this moment.
k_{pc}	The key partition of the child node of the base station
k_{pgc}	The key partition of the node which is after two hops of the base station
k_{path}	The session key of the base station and the node. The session key is the combination of all key partitions of the path

A. Algorithm

Network development stage:

System parameters:

- 1) The base station generates a large prime number p and a key k randomly, and then distributes the key into n parts (n is the number of nodes of the network).

$$k_{pn} = k \cdot (k_{p1} \cdot k_{p2} \dots k_{pn-1})^{-1} \mod p$$

- 2) The base station transmits $n-1$ key partitions to each node of the Network.

Network implementation stage:

- 3) When a user executes the query, packet will record the ID of nodes in the path. If a node conform the query, it will return a acknowledge

(ACK) to the base station.

- 4) The base station computes the part of path key k_{part} and transmits it to the node which is conformed to the query.

$$k_{part} = (k_{p1} \cdot k_{ps} \cdot k_{pgs} \dots) \mod p$$
- 5) The node computes the new path key

$$k_{path} = (k_{pn} \cdot k_{p1} \cdot k_{pc} \cdot k_{pgc} \dots) \mod p$$
and the key using in symmetric encryption.
- 6) The node executes hash function to the data, and then uses the key k_{path} to encrypt the data and result of hash function. Finally, the node transmits the encrypted data to the base station.
- 7) The base station uses key partitions of all nodes that it stored at development stage to computes the key and decrypts the encrypted data.
- 8) The base station executes the hash function to the data and compares with the result of packet. If they are equal, the base station will return the data to the client.

B. Secure Requirements and Analysis

In order to prevent malicious attacks in WSN, it should conform to these conditions as mention below [13][18][19].

We consider energy and computation capability, so we use symmetric encryption to encrypt data, and then execute hash function to confirm data confidentiality, data integrity, authentication and availability.

1) Data confidentiality: To confirm the data cannot be read except the sender and the receiver.

We use implicit security to generate key partitions and recreate these key partitions to a session key between the base station and the node. We use this session key to do symmetric encryption.

2) Data integrity: To confirm the data cannot be modified before sending to the receiver.

The attackers can modify the packet even they do not know the content of packet. For this reason, it is not inadequate if we only condition the Data confidentiality. We join the hash function [20] to verify the data, and then ensure the data which is transmitted by sender and received by receiver are the same.

3) Authentication and Availability: To confirm the source of data is a legitimate node, the network is survival and the data is availability.

The base station knows the topology of the network and it record key partitions of all nodes. The base station can broadcast a random number in a time interval, and receives the result of all nodes that the result is computing by the random number and nodes own key partition. The base station computes the result and compare with the result of it received. It can ensure the node is not a forgery node and observe the survival status of nodes of the network.

Our scheme has higher security level than single network-wide key scheme and random key pre-distribution scheme, due to every session key between two nodes are

unique. It can avoid different communication links has same session key. Furthermore, storage space of number of key in our scheme is smaller than fully pair-wise shared key scheme. As shown in Table 2, k is the number of sensor nodes. Every node except base station only store one key partition, so our scheme is suit for WSN.

Table 2. The Comparison of key distribution schemes

Key distribution scheme	Number of keys stored in each node
Single network-wide key scheme	1
Fully pair-wise shared key scheme	$k - 1$
Random key pre-distribution scheme	Depend on the system
Our scheme	1 key partition

C. Complexity Analysis

In this paper, we propose a new key pre-distribution scheme that uses implicit security to process our keys for to key partitions and make the transmission be safe. We need k multiplications and one inversion operation modulo prime p to partition the key and k multiplications to reconstruct the key. Hence the complexity of key distribution and key reconstruct is $O(k)$, k is the number of sensor nodes. Furthermore, our scheme is suit for WSN especially in distribution of large number of nodes. Due to most of operations to build session key is computed by the base station, that it will reduce the overhead of nodes.

IV. CONCLUSION

Because of the innate limitations of sensor nodes, select an efficient mechanism of key distribution will be very important. In this paper, we find the root of a finite field polynomial to achieve the implicit security. We partition the key and distribute to the nodes of network. We can use these key partitions to revert to the original key. Due to most of operations to build session key is computed by the base station, that the scheme will reduce the overhead of nodes and extend survival time of the network. Because of every two nodes in the communication produce different session key, our scheme is securer than single network-wide key scheme and random key pre-distribution scheme. In addition, the storage space to store key of a node is less than the fully pairwise-shared Key, so the scheme is appropriate for WSN.

ACKNOWLEDGMENT

The paper is partially supported by NSC Taiwan under grant 98-2410-H-002-060-MY2.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no.8, pp. 102-114, Aug. 2002.

[2] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor Network Security: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 2, 2009.

[3] X. Ren, and H. Yu, "Security Mechanisms for Wireless Sensor Networks," *International Journal of Computer Science and Network Security (IJCSNS)*, vol.6, no.3, pp.155-161, Mar. 2006.

[4] M. J. C. M. Belinda, and C. S. G. Dhas, "A Study of Security in Wireless Sensor Network," *MASAU Journal of Reviews and Surveys (MJRS)*, vol. 1, issue 1, pp. 91-95, Sep. 2009.

[5] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," *Federal Information Processing Standard (FIPS) Publication 197*, Nov. 2001.

[6] National Institute of Standards and Technology, "Data Encryption Standard (DES)," *Federal Information Processing Standard (FIPS) Publication 46-3*, Oct. 1999.

[7] J. W. Jiang and J. H. Liu, "Research on Key Management Scheme for WSN based on Elliptic Curve Cryptosystem" in *Proc. 1st Int. Conf. Networked Digital Technologies (NDT)*, pp.536-540, Ostrava, The Czech Republic, Jul. 2009.

[8] A. Price and K. Kosaka, "A Key Pre-Distribution Scheme for Wireless Sensor Networks," in *Proc. Wireless Telecommunications Symposium (WTS)*, pp.253-260, Pomona, USA, Apr. 2005.

[9] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[10] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks," in *Proc. IEEE Pervasive Computing and Communication (PerCom)*, pp. 324-328, Kauai, Hawaii, Mar. 2005.

[11] F. Amin, and A. H. Jahangir, and H. Rasifard, "Analysis of Public-Key Cryptography for Wireless Sensor Networks Security," in *Proc. World Academy of Science, Engineering and Technology (WASET)*, vol. 41, pp. 529-534, Montreal, Canada, May 2008.

[12] L. Eschenauer and V. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," in *Proc. 9th ACM Conf. Computer and Communications Security (CCS)*, pp. 41-47, Washington, DC, USA, Nov. 2002.

[13] S. A. Campete, and B. Yener, "Key Distribution Mechanisms for Wireless Sensor Networks: a Survey," Rensselaer Polytechnic Institute, Computer Science Department, TR-05-07, Mar. 2005.

[14] H. Chan, A. Perrig, and D. Song, "Key Distribution Techniques for Sensor Networks," *Wireless Sensor Networks*, ch.1, pp. 277-303, 2004.

[15] H. Chan, V. D. Gligor, A. Perrig, and G. Muralidharan, "On the Distribution and Revocation of Cryptographic Keys in Sensor Networks," *IEEE Trans. Dependable and Secure Computing*, vol. 2, no. 3, pp. 233-247, Jul.-Sep. 2005.

[16] Y. Cheng, and D. P. Agrawal, "Efficient Pairwise Key Establishment and Management in Static Wireless Sensor Networks," in *Proc. Mobile Ad hoc and Sensor System Conference (MASS)*, pp. 550, Washington, DC, USA, Nov. 2005.

[17] A. Parakh and S. Kak, "Online data storage using implicit security," *Information Science*, vol. 179, no. 19, pp.3323-3331, Sep. 2009.

[18] T. Zia, and A. Zomaya, "Security Issues in Wireless Sensor Networks," in *Proc. International Conference on System and Network Communication (ICSNC)*, pp. 40, Tahiti, French Polynesia, Oct.-Nov. 2006.

[19] Y. Qian, K. Lu, and D. Tipper, "Toward Survivable and Secure Wireless Sensor Networks," in *Proc. Performance, Computing, and Communication Conference (IPCCC)*, pp. 442-448, New Orleans, Louisiana, USA, Apr. 2007.

[20] S. Lee, H. Kim, and K. Chung, "Hash-based Secure Sensor Network Programming Method Without Public Key Cryptography," in *Proc. Workshop on World-Sensor-Web (WSW) at International Conference on Embedded Networked Sensor Systems (SenSys)*, Colorado, USA, Oct. 2006.