International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 23231-23241

http://www.ripublication.com

Hashed Identity Based Secure Key and Data Exchange In Wireless Sensor Networks Using IEEE 802.15.4 Standard

K.Lakshmanarao[1], Hima Bindu Maringanti[2]
[1]*CSE-Department, GMRIT, Rajam, 532127, India.Email:lakshmanarao.k@gmrit.org.*
[2]*Department of Computer Applications, North Orissa University, Baripada,757003, Mayurbhanj, Odisha, India.Email:profhbnou2012@gmail.com.*

Abstract

The Communication among wireless sensor nodes are vulnerable to various security attacks, such as, eavesdropping, insertion of malicious nodes, node capture attack and so on. Hence, wireless sensor networks (WSNs) require efficient methods to reduce security vulnerabilities. The efficient key management scheme for WSNs may reduce security threats. This paper proposes a novel, light-weight key management approach HISKDE(hashed identity based secure key and data exchange) to provide secure authentication, data and key exchange. The HISKDE approach is different from the previous key management approaches, since it assigns a unique String-ID to each sensor node. The sensor node-id is used to refer to the corresponding String-ID of the sensor node. The proposed approach generates a random key, uses spiral arrangement to extract a key value for the generated random key and it uses hashed value of a String-ID to provide secure authentication, secret key and data encryption. The proposed method ensures necessary security metrics, reduce storage, communication and computation overheads. The simulation environment uses IEEE802.15.4 MAC standard to reduce the energy consumed by each sensor node. The security analysis and simulations are provided to illustrate the efficiency of the proposed key management approach. The security analysis shows that, the proposed method has a better performance than the existing hashed random key distribution schemes for WSNs, in terms of authentication, node revocation and node capture problem.

Keywords: Random key pre-distribution for WSNs, Hashed-ID based secure authentication, key and data exchange, IEEE802.15.4 MAC standard.

Introduction

A wireless sensor network can be viewed as aspecial category of ad hoc networks consisting of a large number of small, low cost and low power sensor nodes having

the ability to sense the environment, data processing and equipped with wireless data communication components[1-4]. The sensor nodes are highly constrained on resources, viz., storage capacity, battery power, computational speed and bandwidth.

Wireless sensor networks used in many applications like surveillance, military action, home security, weather monitoring, traffic control, health care and so on[1,3]. Applications like military action, battlefield surveillance and some other applications require more secure communication. However, wireless sensor nodes are vulnerable to various attacks such as impersonating, masquerading, physical protection and unattended deployment[1]. Thus, incorporation of security in sensor networks is highly important.

The design of effective key management methods ensures secure communication in wireless sensor networks. Ensuring security metrics[3,5] like node revocation, resilience, collision resistance, forward and backward secrecy is a challenging task for designing an efficient and appropriate key management scheme in wireless sensor networks.

In addition to the security metrics[3,5], the following are the other challenging issues[6] for designing an effective key management method in wireless sensor networks.First, communication between sensors through wireless media can be eavesdropped easily.Secondly, the limited availability of resources like energy, computational capacity and available memory space. Hence, the proposed key management method for resource utilization should be as less as possible.

There are several key management protocols[1,3,5] for wireless sensor networks, that have been proposed. Among those, some of the proposed key management protocols are considered to be feasible, however these protocols have some disadvantages. The Network-Wide Key approach [3] is under serious security threat from node capture attack, as the physical capture of one node would disclose the common/master key which compromises communication of all other nodes in the network. In pair-wise key distribution scheme [1-3] each sensor node maintains many keys in its memory, thus it creates great memory overhead and the communication overhead is also more in order to establish pair-wise keys between neighboring nodes.

In general, most of the ID-Based [1] key agreement schemes are based on public key cryptography [7], however, subjected to the following defects. First, the memory overhead is more since every sensor node should maintain public keys of all other sensor nodes. Second, the encryption /decryption computational overhead is more.

The Hashed random key pre-distribution key management scheme under probabilistic[3] approach is subject to the following defects: there must be a common shared key to establish connection between pair of sensor nodes, in key pre-distribution phase the size of the key pool is very less, the shared-key discovery phase is a time consuming task since each node should select a shared key from the given key pool and the path-key establishment communication delay is more as described in section 2.1 as well as consume more time to establish a path between two nodes when the nodes do not have a common shared key.

To overcome the above mentioned problems as well as to avoid node capture problem and to ensure security metrics [3,5], we propose a novel HISKDE hashed identity based secure key and data exchange scheme for WSNs. It is an alternative

key management approach to direct key establishment in random manner using key generation rules and extracts secret key value from random key using spiral[9] arrangement. The proposed method assigns a unique string-ID to the respective node-Id and calculates the hash value of the unique string-ID of respective sensor node to provide secure authentication, secure data transmission and secure key exchange. The proposed method does not need any master key unlike other key management schemes. The proposed method does not take much time for deployment of sensor nodes, since it does have phases as proposed in RKP-H [2]. Hence, HISKDE reduces deployment time and storage overhead of each sensor node because nodes do not store any security credentials except node identity.

The rest of this paper is organized as follows. We introduce the motivation behind the proposed scheme, the notations used in proposed scheme and different phases in implementing proposed scheme in Section 2. In Section 3, we describe random key generation process, explain extraction of key value from the generated random key and describe hashing of key and the idea of hash function. Section 4 describes encryption of key and data. Section 5 describes the decryption of received key and cipher-text. In Section 6, we provide a detailed theoretical security analysis and comparison with other key management protocols. The simulation environment is described in Section 7, with conclusion in the last Section.

The Proposed Scheme

This section gives details about the main motivation behind development of our scheme, describes the notations used and different phases in the proposed scheme. In general, the sensor networks are classified into two types[3]: hierarchical(HWSN) and distributed(DWSN) as shown in figure 1. In HWSN each node has a dedicated role based on its capabilities: cluster head, sensor node and base station. In distributed WSN, there are no dedicated roles for each sensor node. That is, each node may act as either normal sensor node or cluster head or base station. Hence, the duties of all sensor nodes are of similar kind and communication may occur between any pair of neighbors. In this paper, we mainly consider the wireless sensor networks as distributed WSN.

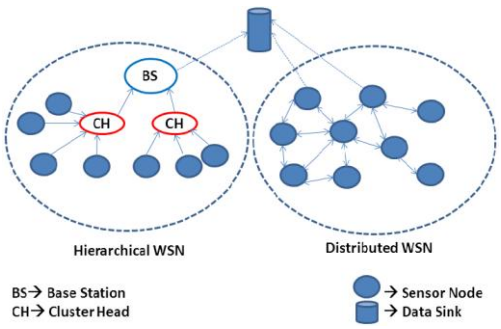

Figure 1: The DWSN and HWSN

Motivation

The random key pre-distribution scheme[2,8] has a key pool which consists of a large number of keys, each key in key pool having a unique key-id. Random distribution of key-ids to each sensor node to establish a shared key between neighbor sensor nodes, does not serve the security in the network. In this process, an adversary may record the key-ids and later on he/she may get connected to the other uncompromised nodes in the network by capturing a couple of nodes in the network. As a result, the key pre-distribution process is not at all secured.

The path key establishment phase [2] consumes more communication overhead to establish a connection between neighboring sensor nodes when they do not have a common/shared key. The (p, q) denote that nodes p and q are neighbors and they are connected through the shared key. When two neighboring sensor nodes x and y does not have shared/link key and want to establish a secure path, consider a pair of neighboring sensor nodes consisting of a shared key to connect node x and y. For example the path $(x, w_1), (w_1, w_2), (w_2, w_3) \dots \dots (w_{n-1}, w_n), (w_n, y)$ is found in between x and y. Hereafter, all the messages between x and y are transmitted through this path.

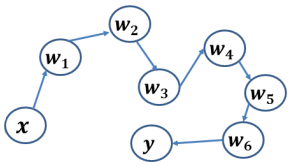

Figure 2: Shows a path between (x, y) when they do not have a link key

In the above figure 2 the nodes x and y are very near to each other but, the communicating path distance is more due to lack of shared key between (x, y). Hence, the above scenario requires more computation power to establish a communication path as well as the data transmission delay is more to reach the destination.

Notations:

The following is a list of notations used to describe our proposed scheme:

The notation → Description
N → *The number of sensor nodes required for initial deployment*
fd → *The number of sensor nodes required for future deployment*
K → *The random key*
K_V → *The key value collected through K using spiral arrangement*
E_K → *The encrypted value of random key K*
A → *The ASCII value of each character of transmitted data*
E_D → *The encrypted data*
$H()$ → *The one way hash function*
$Node_i$ → *The* i^{th} *sensor node*
$String - ID_i$ → *unique string – ID of the* i^{th} *sensor node*
$\|$ → *used for concatenation*
H_{SID} → *one way hash of* $String - ID_i$ *of* $node_i$ *at sender side*
H_{KSID} → *one way hash of* $String - ID_i \| K$ *randomkey generated at sender side*
H_{RID} → *The hash of* $String - ID_j$ *of received* j^{th} $node_j$ *at receiver side*
A_{SID} → *ASCII addition of* H_{SID}
H_{RSID} → *one way hash of* $String - ID_j \| PK_j$ *at receiver side*
PK_j → *is a the possible key selected at receiver side*
PK_S → *is a set of possible keys which satisfy equation* -5
such that $PK_j \in PK_S$ *where* $j = 1 \ldots n$
A_{RID} → *is the ASCII addition of* H_{RID}
P → *Plaintext*
C → *is a ASCII value of each character of received ciphertext*

The HISKDE Algorithms

The proposed method has three phases:the initialization phase, the sender phase and lastly, the receiver phase.

The initialization phase: Before deployment of sensor nodes, we should assign a unique string-ID to each node. If the initial deployment requires N number of sensor nodes, then keeping$N + fd$ number of string-IDs in the code part of the each sensor node makes us easy to add new sensor nodes up to fd number of sensor nodes to the same network in future.

The sender phase: The sender sensor node generates a random key K, key value K_V and hash of$K$ as described in section-III and also performs encryption of both keys and transmits data as explained in the section-IV. Now it keeps the encrypted data, hash of key and encrypted key in security packet and sends to the receiver node.

The receiver phase: Upon reception of security packet, the receiver sensor node performs the key and data decryption as described in section-V.

Key Generation

A random number acts as a key and the generation of a random key[9] follows below given rules:

1. The length of a selected random number should be in between two to five only, that is the number of digits in the key is either 2 or 3 or 4 or 5
2. The key must be a prime number.
3. No digit can be zero.

The range of our HISKDE cryptosystem is in between 10 to 99999. In this range we may get more than 1300 keys which satisfy above rules.

Finding Key Value

The proposed method uses a spiral [9] arrangement as shown in below Table-I. The generated key K gives a key value $K_V$using spiral arrangement using following rules:

1. If a key consists of two digits, then first and second digits of a number represent row and column of a spiral arrangement respectively.
2. If a key consists of three digits, then first and sum of last two digits represents row and column of a spiral arrangement respectively.
3. If a key consists of four or five digits then sum of first two digits and sum of last two digits represents row and column of a spiral arrangement.

31	30	29	28	27	26	49
32	13	12	11	10	25	48
33	14	3	2	9	24	47
34	15	4	1	8	23	46
35	16	5	6	7	22	45
36	17	18	19	20	21	44
37	38	39	40	41	42	43

Table-I

The above spiral arrangement is of size 7X7. The value that will be used for data encryption is found from the grid. For the key 27 the value is 48. We can also increase the size of spiral arrangement.

Key Hashing:

The idea of hashing[2] is very much useful to enhance the security of the proposed scheme. The hash function $H(\ )$ has following prerequisite properties:

a) **Output resistance:**
The hash function gives a unique output that is for a given output x', it should be hard to find any x such that $x' = H(x)$

b) **Collision resistance:**
The output of hash function is always not equal for any two different messages$x_1$and x_2 such that $H(x_1) \neq H(x_2)$

The destination sensor node may get more number of possible keys based on received $E_k$during decryption of a key. Therefore, to avoid conflict at receiver side the following method is used:

$$H_{KSID} = H(String - ID_i \| K_i) \quad (1)$$

Where $H_{KSID}$the key hash combination of both random key K_i(which is equal to K) generated at $node_i$ and $String - ID_i$is unique string assigned to $node_i$. The hashed key H_{KSID}placed in the transmitted data packet which avoids conflict in selecting possible key and useful for sensor node authentication at destination node.

Key and Data Encryption

Each sensor node-ID has a unique string-ID. To encrypt the key the string-ID of a respective node-ID will be hashed and ASCII sum of hashed string-ID is calculated as shown below:

$$H_{SID} = H(String - ID_i) \quad (2)$$

Where $H_{SID,}$the hash of $String - ID_i$ of $node_i$,the encryption of a key is as given below:

$$E_K = (A_{SID} * K) mod\ 256 \quad (3)$$

Where $\mathbf{A_{SID}}$ is the individual ASCII value sum of each character of the$H_{SID}$of $node_i$and the E_K is encrypted value of random key K.

Data Encryption:

There are two steps in data encryption process which are executed at the sender side. The first step is the generation of random key as described in the key generation section-I. By using this random key we determine a value $K_V$from the spiral arrangement as specified in section-II. After completion of above steps we use following equation to perform data encryption.

$$E_D = (A + K_V) mod\ 256 \quad (4)$$

Where **A** is ASCII value of each character and $K_V$is a key value determined using K.

The proposed method uses a simple encryption technique to encrypt transmitted data. We can also use any standard symmetric encryption technique to perform data encryption.

Key and Data Decryption

The key decryption process has two steps which are executed at destination side. The first step is to find possible number of random key set PK_Seach key belongs to the set should satisfy the following eq.6. The second step is to find the exact key among the available possible keys as shown below:

The receiver node selects a unique $String - ID_j$ based on ID of a sending $node_j$and calculates hash as given below:

$$H_{RID} = H(String - ID_j) \quad (5)$$

$$E_K = \left(A_{RID} * PK_j\right) mod\ 256 \quad (6)$$

$PK_S$is set of all possible keys where $PK_j \in PK_S$whenPK_jsatisfies condition specified in equation-6 where$j = PK_1$toPK_n. The key hash is found as specified in eq.7 to each possible key which belongs to $PK_S$and compared with the receivedH_{KSID}key hashH_{KSID}as shown below.

$$H_{RSID} = H(String - ID_j \| PK_j)\ where\ j = PK_1\ to\ PK_n \quad (7)$$

$$if(H_{KSID} == H_{RSID}) \quad (8)$$

If eq.8 is successful then $PK_j$is selected as key at receiver side. The eq.7&8 are repeated till the eq.8 is successful.

Data Decryption:

This task has two steps at destination side the first step is to determine the key value

and the second step is to perform decryption. The first step determines the$K_V$by using selected PK_jas described in section-II and the second step perform the data decryption is as follows:

$$P = C - K_V \tag{9}$$

If P is less than zero in eq.9then the following operation is done:

$$P = P + 256 \tag{10}$$

Where C is ASCII value of individual character of received ciphertext.

Security Analysis and Comparison

In order to measure the performance of our proposed method, we consider two activities 1) security metrics[3,5], 2) comparison with RKP-H[2]. The proposed method ensures following security metrics: node authentication, resilience and node revocation. The table 2 compares RKP-H with our scheme in terms of security metrics.

Security Metrics:

a) Node authentication: The HISKDE method ensures authentication to node and avoids conflicts in possible key generation process at destination side. Each sensor node has a respective unique string-ID, the sender, while sending data it calculates key hash as shown in eq.1.Upon receiving data from sender, the receiver calculates the set of possible keys PK_S as given in equation -6 and find hash key as given in eq.7 then, the received H_{KSID}hashed key is compared with $\mathbf{H_{RSID}}$as described in eq.8. If the comparison is unsuccessful as shown in eq.11 for all possible keys belongs to PK_S then the sender node is malicious otherwise it is authentic.

$$if\ (H_{KSID} \neq H_{RSID})\ then\ sender\ is\ malicious\ node \tag{11}$$

b) Resilience: The proposed method ensures high resilience against node capture attack. When a node was captured then the adversary cannot retrieve any user credentials except node-ID. The adversary cannot compromise any other node in the network with the known sensor node-ID since it is used in an indirect way to ensure authentication, secure key exchange and secure data transmission. Hence, the proposed method HISKED ensures high resilience.

c) Node revocation: If the node authentication process is successful, then the sender node is not malicious otherwise it is malicious. The approach of node authentication of the proposed method can easily recognize an adversary from inserting malicious nodes into the network.

Comparison With Other Key Management Schemes:

In this section, we compared our scheme with the existing RKP-H[2]. Table 2 describes comparison of our scheme with the implementation of RKP-H with respect to the generation of key pool, selection of keys, relationship between node-id and key-id and with different phases of RKP-H. The table 3 compares performance of RKP-H [2] with the proposed scheme.

Table 2: Comparison of our scheme with RKP-H[2]

Key management Schemes ⇨ Attributes ⇩	**RKP-H[2]**	**Our Scheme**
Key Pool	Unstructured	Not defined
Key Selection	Random without replacement	There is no keep pool but the key is selected randomly without replacement.
Node-id & Key-id relation	No relation between Node-id and Key-ids.	The Node-id usages a unique String-id, key is generated randomly and used H(String-id) to secure the key but, key-ids are not defined.
Key pre-distribution	A key pool consist a large number of keys and each key have a unique key-id.	Key pool usage is time consumption. Hence, it generates direct random key.
Shared key discovery	Each node selects a key-id from key pool. Selected key-id used for connectivity between neighbouring nodes.	Shared keys are not used.
Path key establishment	Selected key-id is used for path establish when the neighbours do not have a shared key	This is not required and it is time consuming process as shown in figure 2.

Table 3: Comparison of performance of our scheme with RKP-H[2]

Key management Schemes ⇨ Attributes ⇩	**RKP-H[2]**	**Our Scheme**
Storage overhead	N keys+N key-ids	N keys only
Communication overhead	K_i+h+transmitted data	E_K+keyhash+ transmitted data
Computational overhead	K_i+H+h	$K+K_V+ H_{KSID}$
Resilience against node capturer problem	Poor	High
Node authentication	Poor	High
Node Revocation	Poor	High

K_i(key with key id i)+H(hash function)+h(number of times that a key is hashed)

Simulation

In addition to the security analysis, we perform experimental simulation using NS2 with mannasim. The simulation uses IEEE 802.15.4 [10-12] standard to reduce power consumption of each sensor node. In this experiment the number of sensor nodes is 25 and the environment range is 500X500. The figure 3.a shows the security packet transmission between sensor node-4 and sensor node-2. The figure 3.b shows the simulation approach between sensor node-4 and sensor-2.

Figure 3: (a) packet transmission between node-4 and 2 **(b)** simulation result between node-4 and 2.

Conclusion

This paper proposes HISKDE as a light weight key management with secure authentication and data transmission. It focuses on ensuring necessary security metrics for WSNs. The proposed method tries to avoid node capture problem, communication overhead, computation overhead, storage overhead and provides effective node authentication. Each sensor node only stores node-id. The results of simulation match the security analysis.

References:

[1] Junqi Zhang and Vijay varadharajan, "Wireless sensor network key management survey and taxonomy", Journal of Networks and Computer Applications(ELSEVIER) 33(2010)63-75

[2] Tzu-Hsuan Shan and Chuan-Ming Liu, "Enhancing the Key Pre-distribution Scheme on Wireless Sensor Networks", IEEE Asia-Pacific Services Computing Conference, 2008.

[3] Paulo S.L.M. Barreto et al,"A survey on key management mechanisms for distributed Wireless Sensor Networks", Computer Networks Elsevier,Vol. 54,PP. 2591–2612, 2010.

[4] Marcos A. Simplicio Jr et al, "Survey and comparison of message authentication solutions on Wireless sensor networks", Ad Hoc Networks, Elsevier, Vol. 11,PP. 1221-1236, Aug 2012.

[5] Xiaobing He, et al, "Dynamic Key management in Wireless sensor networks: A survey", Journal of Network and Computer Applications, Elsevier,Vol. 36,PP. 611–622, 2013.

[6] F.Amin, A. H. Jahangir, and H. Rasifard, "Analysis of Public-Key Cryptography for Wireless Sensor Networks Security", World Academy of ENGINEERING AND TECHNOLOGY VOL 17 2008.

[7] Tzu-Hsuan Shan and Chuan-Ming Liu, "Enhancing the Key Pre-distribution Scheme on Wireless Sensor Networks", IEEE, Asia-Pacific Services Computing Conference, 2008.

[8] Ashok Kumar Das, "An Identity-Based Random Key Pre-Distribution Scheme for Direct Key Establishment to Prevent Attacks in Wireless Sensor Networks", International Journal of Network Security, Vol.6, No.2, PP.134–144, Mar. 2008.

[9] Md. Didarul Alam Chawdhury and A.H.M. Ashfak Habib, " Security Enhancement of MD5 Hashed Passwords by Using the Unused Bits of TCP Header", Proceedings of 11th International Conference on Computer and Information Technology (ICCIT 2008) 25-27 December, 2008, Khulna, Bangladesh.

[10] Kyoung-HakJung et al, "An adaptive collision resolution scheme for energy efficient communication in IEEE 802.15.4 networks", Computer Networks,Elsevier, Vol. 58, PP. 39–57, Aug 2013.

[11] Francesca Cuomo et al, "Performance analysis of IEEE 802.15.4 wireless sensor networks: Aninsight into the topology formation process", Computer Networks,Elsevier, Vol. 53, PP. 3057–3075, Aug 2009.

[12] Murat Dener, "Security analysis in Wireless Sensor Networks", Hindawi Publishing Corporation, International Journal of Distributed Sensor Networks Volume 2014, PP.1-9 Oct 2014.