A Dynamic Authentication Scheme for Hierarchical Wireless Sensor Networks*

Junqi Zhang[1,2], Rajan Shankaran[1], Mehmet A. Orgun[1], Abdul Sattar[2], and Vijay Varadharajan[1]

[1] Department of Computing, Macquarie University, Australia
{janson,mehmet,rshankar,vijay}@science.mq.edu.au
[2] Institute for Integrated and Intelligent Systems, Griffith University, Australia
A.Sattar@griffith.edu.au

Abstract. Sensor networks offer economically viable solutions for a wide variety of monitoring applications. In surveillance of critical infrastructure such as airports by sensor networks, security becomes a major concern. To resist against malicious attacks, secure communication between severely resource-constrained sensor nodes is necessary while maintaining scalability and flexibility to topology changes. A robust security solution for such networks must facilitate authentication of sensor nodes and the establishment of secret keys among nodes In this paper, we propose a decentralized authentication and key management framework for hierarchical ad hoc sensor networks. This scheme is light weight and energy aware and reduces the communication overhead.

Keywords: Authentication, Wireless Sensor Networks.

1 Introduction

Ad hoc wireless sensor networks are self organizing wherein all nodes (either moving or stationary) can both provide and relay data. They provide solutions to a range of monitoring problems such as target tracking in battlefields, forest fire detection, medical monitoring and emergency response. However, this dynamic feature of wireless sensor networks poses security challenges which are aggravated not only due to the underlying peculiarities of sensor nodes such as small memories, weak processors, limited energy but also because they are prone to frequent topological changes with the topology being multi hop in nature. A robust security solution for such networks must facilitate authentication of sensor nodes and the establishment of secret keys among nodes.

Traditional authentication frameworks based on public key cryptography [7,18] and PKI [10] are not suitable for WSNs since the sensor network will ultimately consist of small, low-powered devices that are mobile and this necessitates alternatives to authentication based on central authorities and public key certificates. Due to the limited bandwidth and communication being the most expensive operation in terms of energy,

* This research has been supported in part by an Australian Research Council (ARC) Discovery grant (DP0452628) and a Macquarie University Research Development Grant (MQRDG).

P. Sénac, M. Ott, and A. Seneviratne (Eds.): MobiQuitous 2010, LNICST 73, pp. 186–197, 2012.

messages should not be extended significantly in length when applying security services. Apart from this, a security service that is peculiar to sensor networks is Broadcast/group authentication wherein a sending node can broadcast/multicast a message to multiple nodes in an authenticated way. Some schemes such as [17] address this problem but are not scalable as the number of nodes increases.

An orthogonal problem in providing security solutions is to facilitate the provision of a key management infrastructure. Since the sensor nodes suffer from limited memory, battery and processing and communication capabilities traditional key management mechanisms such as those based on asymmetric cryptography are unsuitable for WSNs as they incur high computational overhead. A major limitation of these schemes is that most of them rely on a trusted third party (TTP), thus not fulfilling the self-organization requirement of an ad hoc sensor network. Some solutions based on random key predistribution [4,8,13] impose a limitation on the number of sensor nodes that can be compromised. Once a threshold is crossed, the entire network will be at risk of becoming compromised. Some schemes with a trusted intermediary [3,19] to establish key management have the problem of trusted intermediary being compromised.

In this paper, we address the problem of security by introducing the notion of a hierarchy in network topology wherein we divide the network into clusters, each of which consisting of a small group of nodes. The proposed model is distinguished with low power consumption, less computation workload and enhanced security and equipped with protocols that define how keys are distributed, added, revoked, and updated during the life time of the sensor network.

The remainder of this paper is organized as follows. Section 2 presents a discussion on security issues in hierarchical ad hoc wireless sensor networks. In section 3, we define a framework for authentication and key establishment protocols for such networks. This framework addresses issues that relate to group key generation, distribution and update. In section 4, we propose authentication protocols for both intra and inter cluster environments. In section 5, we compare our framework with similar works. Finally in section 6, we provide concluding remarks.

2 Security Issues in WSN

In this section, we provide an overview of the system constraints, security issues, and the security requirements in wireless sensor networks.

The constraints to be considered in sensor networks include two aspects: the network building (hardware) and network operating (software) [1]. The network infrastructure building aspect involves: infrastructure, deployment (location fixed) and mobility, network topology, density and network size, connectivity, and life time. The two common communication modes are the infrastructure based network and ad hoc network. In infrastructure based wireless sensor networks, the sensor nodes can only communicate directly with base stations. In ad hoc sensor networks, nodes can communicate directly with each other without any infrastructure.

The operating (software) aspects include self configuration, data aggregation and dissemination, node addressability, real time, reliability and security. WSNs must be self-organized to establish a topology to support communication. The sensor nodes will

preclude manual configuration before deployment in networks. The networks are able to continuously and periodically to reconfigure themselves for dynamically changing nodes. Data aggregation is the summarization of the traveling data through the sensor network.

Security is a must for many applications of WSNs. There are a variety of potential attacks that breach security. These threats can be classified into four categories: changing message routing path attacks, injecting message attacks, disclosing message attacks and other attacks. Changing message routing path attacks includes sinkhole attack, wormhole attack, sybil attack, replay attack, selective forwarding attack, and non-replication or impersonation attack. In a sinkhole attack [12], an adversary tries to make all or some traffic from a certain area pass a compromised node. The attacker advertises a high quality link to the base station to change the message routing path. Sinkhole attack can enable other attacks such as privacy attack, and selective forwarding attack. In a wormhole attack, the attacker tunnels the captured data into a private link between two colluding nodes. The data can be dropped, forwarded or modified by malicious nodes [11]. In a replay attack, the attacker retransmits captured messages to disrupt or compromise the network. Without protection, the receiver node cannot distinguish a replayed message from the normal message [12]. In a selective forwarding attack, the malicious node will selectively drop some messages [12].

An inject message attack can be divided into inject false message attacks and injecting extra message attacks. Inject false message attacks include sybil attack, and non-replication or impersonation attack. In the sybil attack, the attacker employs a compromised node to masquerade as many other nodes. This can affect routing, data aggregation, and clustering [16] . The non-replication or impersonation attack is similar to the sybil attack. The difference is that the malicious node masquerades as an already existing node [6], which can lead to corrupted or misrouted data. Injecting extra messages attacks includes denial of service attacks, HELLO flood attacks and so on.

Disclosing message attacks include: traffic analysis attack and privacy attack. In the traffic analysis attack, the attacker locates an important node such as the base station so that it can be made unavailable or compromised. In the privacy attack, the attacker tries to discover the message by monitoring the network traffic and listening to the data.

3 Authentication in WSN

The security requirements for WSNs are similar to other networks. They may include the authentication, integrity, freshness, availability and confidentiality.

Authentication is the process of verifying the identity of someone or something. The three types of cryptographic functions used for authentication are hash functions, secret key functions, and public key functions. In traditional networks, the common way to authenticate someone is the use of public key functions. In WSNs, it is usually assumed that public key cryptography can not be used because of the elaborate constraints. This means that the two communicating entities must use secret key functions and hash functions. In WSNs, there are two types of authentication: device level authentication and group level authentication. The device level authentication means that a message is proved to originate from a certain device, whereas the group level authentication means a message is proved to originate from a certain group of devices.

Several authentication schemes have been proposed for WSNs. These schemes can be divided into three types: public key cryptography based, symmetric keys and hash functions, and one way key chain based on hash functions.

Public key based approaches include those based on the RSA public key cryptosystem and Elliptic curve cryptography. TinyPK uses the lower exponent variant of the RSA public key cryptosystem to implement authentication of an external party [20]. The external party is an entity that wishes to establish secure communication with the sensor network. The private part of the RSA is carried out at the certificate authority (CA). The nodes only need to implement the public parts, i.e., the data encryption and signature verification as this is much faster to perform than the private parts in RSA. The public key based approach can incur high computational overhead and network bandwidth consumption. Elliptic curve cryptography (ECC) can be implemented with a much smaller key size and memory usage than RSA. Blaβ and zotterbart give a software implementation of ECC on an Atme microcontroller [2]. ECC has the computational and memory size advantages, but it suffers from more complex arithmetic primitives and a large number of temporary operands [9].

In private keys and hash functions based schemes [21,23,22], each symmetric authentication key is shared by a set of sensor nodes. If an intruder compromises a sensor node, the shared key will be disclosed. Hence these approaches are not resilient to a large number of node compromises.

In one-way key chain type of schemes, the key hashed key chain and the technique of delayed disclosure of keys are used. μTESLA [17] and its variants [14,15,6] are such approaches. In μTESLA, a key chain with delayed key disclosure is used to create an asymmetry in time among the broadcasting source (sinks or users) and the receiver (sensor node) to emulate public key cryptography. Initially, sensor nodes are preloaded with $K_0 = h^n(x)$, where $h^n()$ is a hash function and x is the secret held by the sink (user). The sender(user or sink) sets up time intervals and in each time interval one key is used. During time interval I_1, $K_1 = h^{n-1}(x)$ is used to generate message authentication code (MAC) for all the broadcast messages sent. During time interval I_2, the sender (sink or user) broadcasts K_1, and sensor nodes verify $h(K_1) = K_0$). With K_1, the sensor nodes can verify the authenticity of the message received during the time interval I_1. The receiving sensor nodes need to verify that the key was not disclosed when it received the message. Therefore, loosely synchronized clocks between the sender (sink) and the sensor nodes are needed.

Recently a hierarchical wireless sensor network security protocol was proposed by [5]. This scheme employs hash functions, hash key chains and symmetric keys. Each sensor and the base station share a secret hash key chain. The sensor encrypts the data and sends it to the cluster head. The cluster head collects the data from the sensor nodes and then retrieves the secret keys from the base station. The cluster head decrypts the encrypted message and then sends these data to the base station. This scheme has several advantages. Firstly, it reduces the storage overhead, as each sensor node only stores three keys. Secondly, it reduces the probability for the guessing attack as the sensor nodes change keys once for each transmission. Finally, it uses two way challenge and response authentication method, so it can prevent replay attacks. However, this scheme has several disadvantages. Firstly, cluster heads can disclose all the secret keys of the

sensor nodes in their cluster. A single compromised cluster head can affect a large number sensor nodes. Secondly, the cluster heads need to retrieve the sensor nodes secret key for every data transfer. This would cause communication overhead. Thirdly, the sensor nodes need to frequently change the secret keys for each time of data collection.

In order to mitigate these disadvantages, we propose a new authentication scheme which is similar to that in [5], but with more security and less computation and communication overhead.

4 Authentication Protocols for WSN

In the hierarchical wireless sensor network model, a wireless sensor network consists of a command node (or a base station), cluster heads and numerous sensor nodes which are grouped into clusters. The clusters of sensors can be formed based on various criteria such as capabilities, location and communication range, and usage of different cluster algorithms and strategies.

Each cluster includes the cluster head (or the cluster leader) and a set of distinct sensors. Each sensor has two main functions: sensing and relaying. Sensors probe their environment and gather data. They then transmit the collected information to the cluster head directly in one hop or by relaying via a multi hop path. Sensors transmit or relay data only via short-haul radio communication. A cluster head is in charge of its cluster. It is assumed that each cluster head can reach and control all the sensors in the cluster. Each cluster head receives data from different sensors, and then processes the data to extract relevant information, and sends it to the base station (command node) via long-haul transmission.

In the rest of this section, we first give the notations to be used, and then we describe the basic authentication protocols. We also discuss the the authentication protocols for dynamically moving sensor nodes.

4.1 Notation

The symbols and abbreviations used for the protocols are listed in Table 1. The base station stores the following information: two hash functions $H()$, $G()$, all the cluster head and sensors ID_{cl}, and ID_{si}, a shared secret key with each cluster head and sensors K_{bc_l}, K_{bsi}, and a shared secret group key for each cluster K_l, here $l = 1, ...m$, $i = 1, ..n$ for m clusters and n nodes in the sensor network. Each cluster head stores the following data: two hash functions $H()$, $G()$, all the sensor nodes ID_{si} in its cluster, a shared secret key with the base station K_{bc}, a session key for its cluster group K_{sk}, here $si = 1...p$ for a cluster with p sensor nodes. Each sensor stores the following information: two hash functions $H()$, $G()$, sensors ID_{si}, shared secret key with the base station K_{bc}, and the session key for the cluster group K_{sk}.

4.2 Basic Authentication Protocols

We consider the basic authentication protocols with three scenarios: the base station and cluster head(s), the cluster head and the sensors in the cluster, and the cluster head and a cluster head from a different cluster. In order to reduce the computation overhead, we employ symmetric key functions and hash chain functions in these protocols.

Table 1. Notations used in the protocols

	Symbol	Meaning
Base station	$H(), G()$	hash functions
	$ID_{cl}, (l = 1, \cdots, m)$	cluster ID list
	$K_{bc_l}, (l = 1, \cdots, m)$	shared secret with the cluster head
	K_{sk} , $(sk = 1, \cdots, m)$	session key
	$ID_{si}, (si = 1, \cdots, n)$	sensor ID list
	$K_{bs_i}, (i = 1, \cdots, n)$	shared secret with the cluster head
Cluster head	$H(), G()$	hash functions
	ID_{cl}	cluster ID
	K_{bc},	shared secret with the base station
	K_{sk}	session key
	$ID_{si}, (si = 1, \cdots, p)$	sensor ID list
Sensor node	$H(), G()$	hash functions
	ID_{cl}	cluster ID
	K_{sk}	cession key
	ID_{si}	censor ID
	K_{bs},	shared secret with the base station

Scenario one: the base station and the cluster head – This is a mutual authentication protocol between the base station and the cluster head in each cluster. We employ a hash chain to dynamically change the shared key between them. Hence guessing attacks can be prevented. The standard mutual authentication protocol can mitigate the reflection attacks. The authentication transfer protocol for the base station and the cluster head is shown in detail in Figure 1.

There are seven steps in our authentication protocol. The first three steps are for the cluster head to authenticate the base station and the next three steps are for the base station to authenticate the cluster head. We describe them as follows.

$Step$ 1. The cluster head sends the join message with its identity ID_{cl} and p_c (encryption of a nonce N_c and IC_cl) to the base station.

$Step$ 2. Upon receiving the message, the base station decrypts it, then update the shared key by rehashing it $K_{bc1} = H(K_{bc})$, and and then encrypts the nonce using the new shared secret key with this cluster $P_1 = E_{K_{bc1}}(R)$. Then the base station sends P_1 to the cluster head

$Step$ 3. The cluster head decrypts the message received from the base station, and then compares it with the original random number to verify the base station.

$Step$ 4. The base station chooses a nonce N_b, encrypts it with the shared key and sends it to the cluster head. The base station updates the shared key by rehashing it.

$Step$ 5. The cluster head decrypts P_b, and then encrypts the nonce N_b with the updated shared key with the base station $P_2 = E_{K_{bc2}}(N_b)$, and then sends P_2 to the base station along with its identity ID_{cl}.

$Step$ 6. The base station decrypts p_3 with the shared key, and then compares the nonce with the original one. If they are the same, this means the cluster head is authenticated. Then the base station chooses the session key for the cluster group and encrypts it with the dynamical shared secret key $P_3 = E_{k_{bc2}}(K_{sk})$. At the same time, the base

Base station – $H(), K_{bc}$		Cluster head – $H(), K_{bc}$
		(1) Choose nonce N_c $P_c = E_{K_{bc}}(N_c \| \| ID_{cl})$
(2) $D_{K_{bc}}(P_c)$ $K_{bc1} = H(K_{bc})$ $P_1 = E_{K_{bc1}}(N_c)$	$\xleftarrow{Join, ID_{cl}, P_c}$	$K_{bc1} = H(K_{bc})$
	$\xrightarrow{P_1}$	(3)Verify N_c
(4) Choose N_b $P_b = E_{K_{bc1}}(N_b)$ $K_{bc_2} = H(K_{bc_1})$	$\xrightarrow{P_b}$	(5) $D_{K_{bc1}}(P_b)$ $K_{bc2} = H(K_{bc1})$ $P_2 = E_{k_{bc2}}(N_b \| \| ID_{cl})$
	$\xleftarrow{P_2, ID_{cl}}$	
(6) Verify N_b if successful $P_3 = E_{K_{bc_2}}(K_{sk})$ $K_{bc_3} = H(K_{bc_2})$		
	$\xrightarrow{P_3}$	(7) $K_{sk} = D_{K_{bc0}}(P_3)$ $K_{bc_3} = H(K_{bc_2})$

Fig. 1. The Authentication Protocol for the base station and the cluster head

station hashes the shared key $K_{bc3} = H(K_{bc2})$ and saves it for authentication next time. The base station then sends p_3 to the cluster head.

$Step$ 7. The cluster head decrypts p_3 and then obtains the session key for the cluster group. Meanwhile the cluster hashes the shared key $K_{bc3} = H(K_{bc2})$ to achieve the dynamical shared key and save it for next time authentication.

Scenario Two: the base station and a sensor node – In our model, the base station shares a secret key with each sensor node, and the cluster head does not have a shared key with the sensor node. Therefore, the base station and the sensor node need to do a mutual authentication. Then the base station distributes the group key to the sensor. As the cluster head also has the group key for the same with all the sensors in its cluster, the cluster head then shares a group session key with all the nodes in the cluster.

The authentication protocol between the base station and the sensor node is similar to the one between the base station and the cluster head; we omit the details. The difference is that all the communication passes through the cluster head. The authentication transfer protocol for the base station and a sensor node is shown in Figure 2.

Scenario Three: two cluster heads – The authentication protocol between two cluster heads is similar to the mediated authentication with KDC (Key Distribution Center). The base station acts as a key distribution center. First, it generates the shared key for the two cluster heads, and then the two cluster heads mutually authenticate each other using this shared key. The protocol is shown in Figure 3. The steps in this protocol are described as follows.

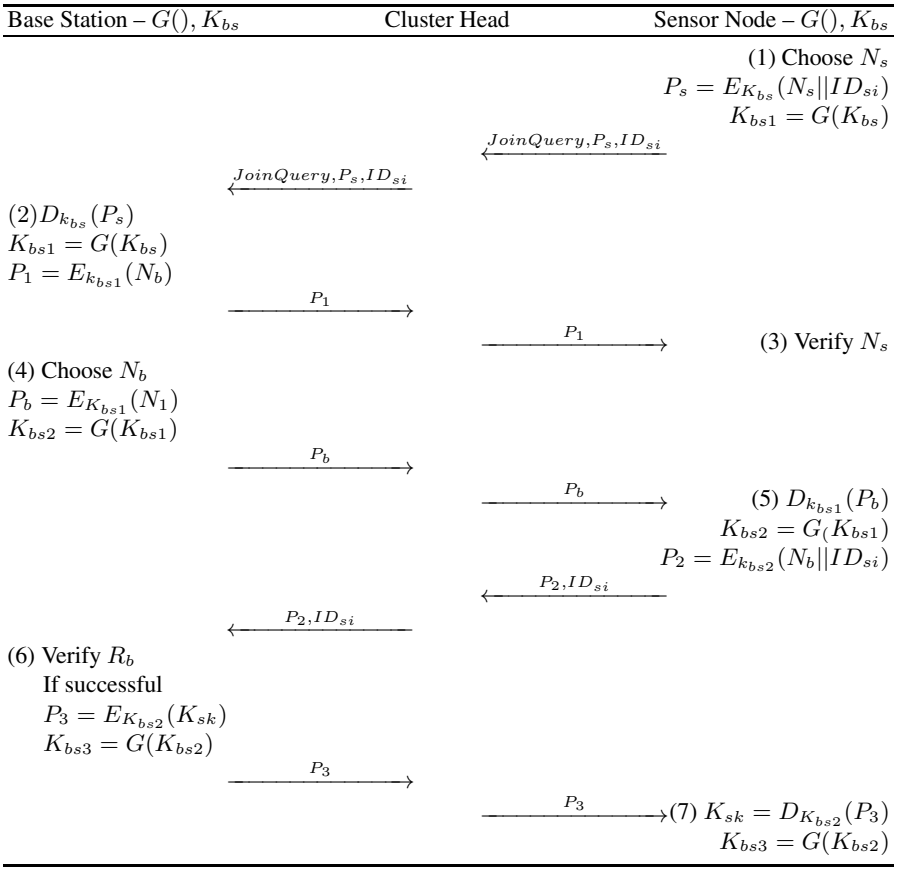

Fig. 2. Authentication Protocol for cluster head and Sensor Node

Step 1. The cluster head A sends the request for communicating with the cluster head B to the base station.

Step 2. The base station creates the session key K_{AB} shared by the cluster head A and the cluster head B. It then generates the ticket $ticket = E_{CB}(K_{AB})$, and encrypts the session key with the shared key with the cluster head A $F_0 = E_{CA}(K_{AB})$. The base station sends F_0 and $ticket$ to the cluster head A.

Step 3. The cluster head A decrypts the F_0 and obtains the session key K_{AB}, and then chooses a random number R_1. The cluster head A sends R_1 and the tickets to cluster head B

Step 4. The cluster head B decrypts the ticket with the shared secret key with the base station and obtains the shared session key K_{AB}. Then the cluster head B encrypts the random number R_1 with the shared secret key K_{AB} $F_1 = f(K_{AB}, R_!)$. The cluster head B sends F_1 to the cluster head A.

Step 5. The cluster head A decrypts F_1 and verifies the random number R_1.

Step 6. The cluster head B chooses a random number R_2 and sends it to the cluster head A.

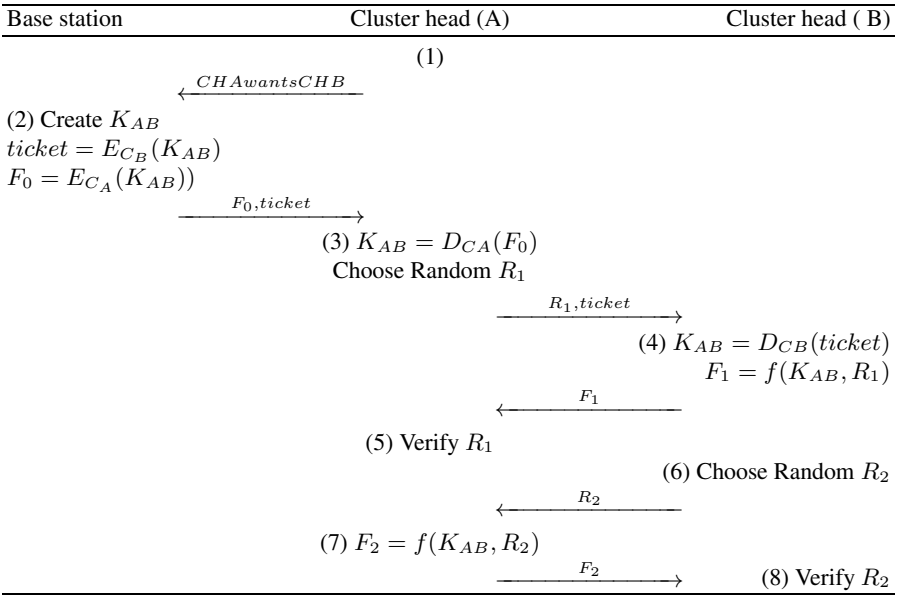

Fig. 3. Authentication Protocol for two cluster heads

Step 7. The cluster head A encrypts the random number R_2 with the shared session key K_{AB} $F_2 = f(K_{AB}, R_2)$, and then sends F_2 to the cluster head B.

Step 8. The cluster head B decrypts F_2 and verifies the random number R_2.

4.3 Authentication Protocols for Dynamical Movement of Nodes

In this section, we summarize the authentication protocols for the following three scenarios of dynamically moving nodes: (1) a sensor node moves from one cluster to another cluster, (2) one cluster is partitioned, and (3) two clusters merge.

Scenario One: A Node Moves from One Cluster to Another – A sensor node moves from cluster A to cluster B. There are two cases to consider: (1) an existing node leaves its cluster, and (2) an existing sensor node joins a new cluster. When one existing sensor node leaves a cluster, if the cluster does not want forward secrecy, then there is nothing to do. If the cluster wants forward secrecy, the new cluster group key must be redistributed. For the case that an existing sensor node joins a cluster, authentication can be done through the cluster head as the sensor nodes still share the dynamical secret key with the base station. If there is no requirement for backward secrecy, the cluster just sends the cluster group session key to the newly joined sensor node; otherwise, the sensor nodes in the cluster need a new session key to be distributed among them.

Scenario Two: Cluster partitioning – There are two cases for cluster partitioning: (1) some of the sensor nodes leave the cluster and organize a new cluster with a new cluster head, and (2) the sensor nodes are divided into two new clusters with two new cluster

heads. In case one, the original part of the group only needs to distribute a new session key. The new cluster head needs to be authenticated, and then all the sensor nodes with the new cluster head form a new cluster and a new session key is generated. In case two, both of the new cluster heads need to be authenticated, and then all the sensor nodes within each new cluster form two new clusters and a new session key is also needed.

Scenario Three: Clusters merging – Cluster merging may include two cases: (1) the sensor nodes merge into an existing cluster, and (2) the sensor nodes merge with a new cluster head. In case one, the existing sensor nodes are authenticated through the cluster head, and a new session key needs to be distributed to all the sensor nodes in the cluster. In the second case, the new cluster head needs to be authenticated, and then all the sensor nodes form a new cluster; a new group cluster session key also needs to be distributed to all the sensor nodes.

5 Discussion and Analysis

In this section, we compare our proposed scheme with the DSKG scheme as both of them have the similar hierarchical architecture. We compare them on several aspects: communication overhead, memory overhead, computation overhead and security etc. The comparison table is shown in table 2

Table 2. Comparison of the DSKG scheme and the new scheme

	DSKG scheme	New scheme
Communication between base station and cluster head	much more	less
Communication for cluster heads and sensor node	roughly same	roughly same
Memory overhead for nodes	roughly same	roughly same
Memory overhead for cluster head	more	less
Cryptographic functions	hash & symmetric	hash & symmetric
Computation overhead for node	roughly same	roughly same
Computation overhead for cluster head	more	less

Communication overhead – For DSKG, communication is not efficient for several reasons. Authentication is required for each data transition. The cluster head needs to request the secret key from the base station for each message it obtains from the sensor node. In our new scheme, we use a group key, so there is no need to transfer the secret key for each message.

Memory overhead – In DSKG, a cluster head needs to store more data than our scheme because it stores lots of messages before it transmits them to the base station. In our scheme, the cluster head does not store much data. For sensor nodes, memory overhead is roughly the same.

Computation overhead – For DSKG, there are more authentication processes and more encryptions and decryptions in the cluster head. Hence there is more computation overhead than our new scheme.

Security – Our scheme employs the dynamical hash key chain technique and has the same advantages over the DSKG scheme. Firstly, it reduces the probability for guessing attacks as the sensor nodes exchange keys once for each authentication. Secondly, it uses two way challenge and response authentication method, so it can prevent reflection and replay attacks. One drawback of our scheme is that if the sensor nodes in one cluster change frequently, the group key will have to be changed. Therefore, our scheme will be have a better performance if it is applied to the relatively less changing clusters. In the real world, most applications may fall under this category.

6 Concluding Remarks

Wireless Sensor networks provide economically viable solutions for a wide variety of monitoring applications. When WSNs are deployed in an unattended or hostile environment, security is a major concern. In this paper, we analyzed WSNs security issues and classified them into three categories. We also reviewed the proposed authentication approaches . We proposed a dynamical key authentication scheme for hierarchical WSNs. This new scheme has several advantages over a recently proposed similar scheme.

References

1. Belinda, M.J.C.M., Dhas, C.S.G.: A study of security in wireless sensor networks. MASAUM Journal Of Reviews and Surveys 1, 91–95 (2009)
2. Blab, E.O., Zitterbart, M.: Towards acceptable public key cryptography in sensor networks. In: The 2nd International Workshop on Ubiquitous Computing (2005)
3. Chan, H., Perrig, A.: Pike: Peer intermediaries for key establishment in sensor networks. In: Proceedings of IEEE Infocom. IEEE Computer Society Press, Los Alamitos (2005)
4. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy, p. 197. IEEE Computer Society, Washington, DC (2003)
5. Chen, C., Li, C.: Dynamic session key generation for wireless sensor networks. EURASIP Journal on Wireless Communications and Networking (2008)
6. Deng, J., Han, R., Mishra, S.: Countermeasures against traffic analysis attacks in wireless sensor networks. In: Proceedings of 1st IEEE Conference on Security and Privacy for Emerging Areas in Communication Networks (2005)
7. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22, 644–654 (1976)
8. Du, W., Deng, J., Han, Y.S., Varshney, P.K.: A pairwise key pre-distribution scheme for wireless sensor networks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 42–51. ACM Press, New York (2003)
9. Gaubatz, G., Kaps, J.P., Sunar, B.: Public key cryptography in sensor networks - revisited. In: Castelluccia, C., Hartenstein, H., Paar, C., Westhoff, D. (eds.) ESAS 2004. LNCS, vol. 3313, pp. 2–18. Springer, Heidelberg (2005)
10. The Open Group. Architecture for Public-Key Infrastructure, APKI (1999)
11. Hu, Y.C., Perrig, A., Johnson, D.B.: Packet leashes: a defense against wormhole attacks in wireless networks. In: Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1976–1986. IEEE, Los Alamitos (2003)

12. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: Attacks and countermeasures. In: Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, AK, USA (2003)
13. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 52–61. ACM Press, New York (2003)
14. Liu, D., Ning, P.: Multi-level mtesla: Broadcast authentication for distributed sensor networks. ACM Transactions in Embedded Computing Systems (TECS) 3 (2004)
15. Zhu, S., Liu, S.J.D., Ning, P.: Practical broadcast authentication in sensor networks. In: Proceedings of Proc. of MobiQuitous, Mobicom 2001 (July 2005)
16. Newsome, J., Shi, E., Song, D., Perrig, A.: The sybil attack in sensor networks: Analysis & defenses. In: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, pp. 259–268 (2004)
17. Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E.: Spins: Security protocols for sensor networks. In: Proceedings of 7th Annual ACM International Conference on Mobile Computing and Networks (Mobicom 2001), Rome, Italy (2001)
18. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 22, 120–126 (1978)
19. Singh, K., Muthukkumarasamy, V.: A minimal protocol for authenticated key distribution in wireless sensor networks. In: Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing, Bangalore, India (December 2006)
20. Watro, R., Kong, D., Cuti, S.F., Gardiner, C., Lynn, C., Kruus, P.: Tinypk: Securing sensor networks with public key technology. In: Proceedings of the 2nd ACM Workshop on Security of Ad hoc and Sensor Networks, Washington DC, USA (2004)
21. Ye, F., Luo, H., Lu, S., Zhang, L.: Statistical en-route filtering of injected false data in sensor networks. In: IEEE Infocom 2004 (March 2004)
22. Zhu, S., Setia, S., Jajodia, S.: Leap: Efficient security mechanism for large-scale distributed sensor networks. In: Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS), Washinton DC, USA (2004)
23. Zhu, S., Setia, S., Jajodia, S., Ning, P.: An interleaved hop-by-hop authentication scheme for filtering false data in sensor networks. In: IEEE Symposium on Security and Privacy (2004)