# Lightweight Session key Management scheme in Sensor Networks

Hui Jin, Haiguang Chen
*Mathematic and science college*
*Shanghai Normal University*
*Shanghai, P. R. China, 200234*
*{chhg,jinhui}@shnu.edu.cn*

## Abstract

*The necessity to secure the communication between wireless sensor networks nodes becomes important with the system nodes are being deployed for a wide variety of applications. In this paper, we propose a new approach for session key management among group sensor nodes. We use initial trust node to build a small set of shared keys, low-cost protocols enable neighboring sensors to authenticate and establish secure local links. Each shared key used only once after the set of shared keys are built. Our session scheme exhibits a nice power efficient and excellent updating communications keys. At the same time our scheme updating keys go with exchanging message. Extensive analyses and simulations are conducted to evaluate the proposed schemes, and the results show that the proposed management schemes can achieve a good level of security.*

## 1. Introduction

Distributed wireless sensor networks include large self-organizing nodes having locally communicating system. Those nodes are being deployed for a wide variety of applications [1]. A lot of interest and effort are being focused on this new network topic. It is important to get security in wireless sensor networks (WSNs) and encrypt the communicating messages among these nodes.

But, all the sensors nodes don't have sufficient memory and processing power. The approaches based on public key cryptography or on the Diffie-Hellman key exchange protocol cannot be used in WSNs[2], because the cryptographic schemes are too expensive for the most resource constrained sensor nodes.

Very recently, a number of key management schemes for WSNs have been proposed [3,4,5,6,7,8] to protect the communications. Eschenauer and Gligor proposed a random key pre-distribution scheme: Each node is loaded with a random subset of keys from a large key pool. To agree on a key for communication, two nodes need find one common key within their respective subsets and use that as their secret communication key [4]. The difference between the q-composite scheme and the scheme in [4] is that q common keys ($q \geqslant 1$), instead of just a single one, are needed to establish more secure communication between a pair of nodes. It is shown that by increasing the value of q network resilience against node capture is improved [5].Du et al presented a random key pre-distribution scheme and using deployment knowledge to avoid unnecessary key assignments [8]. Typical those schemes having each node pre-load a large deal of keys to have a reasonable probability of sharing one with a neighbor node, those schemes still require a large amount of memory, as well as an infrastructure to load the keys into the sensor nodes, which isn't necessary for some applications such as medical or environment monitor.

In this paper, we focus on session key management in commodity WSNs. Our scheme without relying on expensive cryptography or trusted center servers, and we don't need any key pre-distribution phase to build the local security link among innocent neighbor nodes.

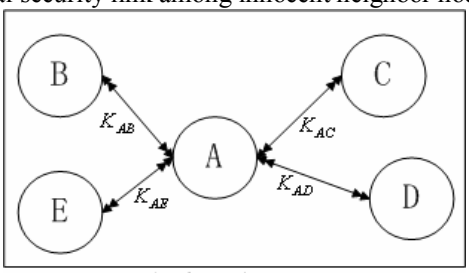


Figure 1:Shared-session Keys

In this paper, we propose a new protocol to construct session key management among WSNs. The session key is also known as shared-session keys or link dependent keys. For different security links we will use different shared-secret keys. As shown in Figure 1, node A has four security link to node B, C, D and E, if node A wants to exchange message with B, it should use key $K_{AB}$ , if node A wants to exchange packets with C, it should use key $K_{AC}$. These two shared-session keys are independent; therefore even if the adversary

captures one of the security link or node, the rest of the networks is still safety. The main contributions of this paper are as follows:

1. We describe a more realistic attack model which used in non-martial communication in WSNs.
2. We present a lightweight session key management scheme for WSNs, which is similar to a One-Time Pad and efficient to those sensor nodes.
3. We analysis the security of session key management for WSNs, and provide new way to trade off security for cost and usability.

## 2. Related work

In this section, we will briefly introduce some related security work in WSNs. Wenliang Du[6] proposed a pair-wise key pre-distribution scheme for WSNs, it based on a symmetric matrix to reduce the storage of the sensor and computation, it has improved the security level of WSNs. But it has a weakness when a number of node has been captured, the whole nodes of WSNs will be captured soon. The logical tree-basiced schemes proposed by wallner et al[10],wong et al[11],and Balenson et al [12] can achieve logarithmic broadcast size, storage, and computational cost. But most of those scheme [3,4,5,6, 7,8,10,11,12] need key pre-distribution phase, and each node must load a great number of shared key to build secure link. It cannot suitable for a realistic attack model. We will discuss the realistic attack model in section 4.1. Almost all these protocols assume secret information is set up before nodes deployed. In this paper we explore key setup without any prior information and without load shared keys before deploying those nodes

## 3. Building blocks

In cryptography, the one-time pad (OTP) is the only theoretically unbreakable method of encryption: the plaintext is combined with a random "pad" the same length as the plaintext. The method was proven the only unbreakable cipher in an information-theoretic sense by Claude Shannon. However, it has drawbacks in practice: it requires perfectly random one-time pads; secure generation and exchange of the one-time pad material, which must be at least as long as the message; and careful treatment to make sure that it is disposed of correctly and never reused. These implementation difficulties have led to examples of one-time pad messages being broken, and are so serious that they have prevented the one-time pad from being adopted as a widespread tool in information security.

### 3.1 An example of one-time pad

We show an example of how to use a one-time pad to encrypt the message $M(x)$.

If node A want to send a message $M(x)=(x_1,x_2,...,x_n)$ to node B, and choose the key $V_0(v)=(v_1,v_2,...,v_n)$ which A and B has known before. So in node A, use the $E_v(M(x))=(x_i+v_i) \bmod N \quad where(i=1,2,3...n)$ to encrypt the message $M(x)$. $N \geq 26$ is an integer. Where we can select $N$ =26. So One-Time Pad is $E_v(M(x)) = ((x_1 + v_1)\%26, (x_2 + v_2)\%26, ...(x_n + v_n)\%26)$ then the cipher text, $E_v(M(x))=(x_1',x_2',x_3',...,x_n')$, to be sent to node B. Node B uses the same process, but in reverse, to obtain the plaintext. Here, $D_v(x') = ((x_1' - v_1)\%26, (x_2' - v_2)\%26, ..., (x_n' - v_n)\%26)$. In this way, we can obtain the plaintext $M(x)$ send by node A using One-Time Pad.

## 4. Implement of session key Scheme

In this section, we present the basic feature of our scheme, deferring its analysis for next section.

Firstly, we suppose a real world attack model for WSNs in section 4.1, and then discuss how to build up a local secure link and how to update the session keys for trusted nodes.

### 4.1 A Realistic attack model

In prior work, most researches assumed that the attacker has high capable and motivated. These nodes can monitor and store the entire exchanged message. It maybe realistic for wired networks, but cannot suitable for WSNs, especially to non-martial application, which for cost reasons have extreme limitations for sensor nodes and also require that the pre-deployment setup must be minimal. We assume our attack model as follows:1)the attacker don't execute any active attacks during the wireless sensor node deployment phase. 2) The attacker is able to monitor only a small part of the communications of WSNs in node deployment phase. After key exchange complete, the attacker can monitor all of the traffic. 3) The attacker cannot have physical access to the deployment site in the node deployment phase, and the node cannot be physical captured. In summary, the attacker cannot active attack during node deployment phase. These hypotheses are realistic because deployment phase is a very brief and the opportunity to be captured is low. So, we focus on

update shared-session keys after the nodes have been deployment and build a security link among nodes.

### 4.2 Key Negotiation

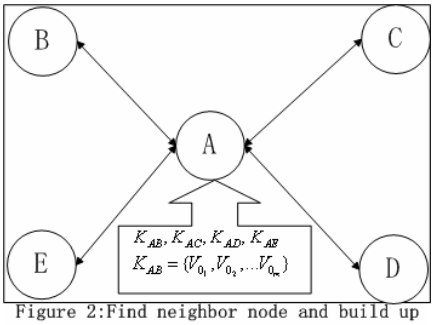

Figure 2:Find neighbor node and build up set of Shared-session Keys

When the sensor node has been deployed, then the node broadcasts its identification, say A. If a neighbor node B hears node A, the node B generates a random number $n_0$ then send it to node A. Both node A and node B will use this random seed number $n_0$ to generate a set of shared-session keys $K_{AB}(V_{0_1}, V_{0_2}, ...V_{0_m})$, and $V_{0_i} = (v_{i1}, v_{i2}, ....v_{in})$, where $i = (1, 2, ...m)$. After this phase the local secure link has been build up, as shown in Figure 2. Then in section 4.3, we will describe how to update the session shared keys.

### 4.3 Communication and re-key scheme

In Section4.2 we know the security links has been build among the neighbor nodes, and each node has a symmetry keys to communicate. If a node has a message $M(x) = (x_1, x_2, ....x_n)$ to send to another node, then the node use the following process to obtain the cryptograph $E_v(M(x)) = ((x_1 + v_{11})\%N, (x_2 + v_{12})\%N, ...(x_n + v_{1n})\%N)$, then send it to the received node. Because the shared key vector $V_{0_1}(v) = (v_{11}, v_{12}, ...v_{1n})$ has been known between sender and receiver nodes, so the receiver use the reverse process and easily get the plaintext $M(x) = (x_1, x_2, ....x_n)$. After the process, both sender and receiver will never use $V_{0_1}(v)$. And will use $V_{0_2}(v)$ for the next communication shared key. In this way, shared keys are only used once. As shown In Figure 3.

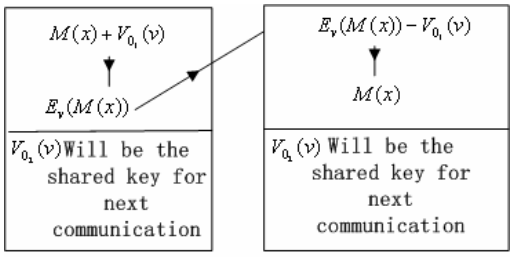

Figure 3:The message exchange phase and use another shared key for next communication

If all of the set of shared session keys has used up, the node will send a random number again and use this seed number to generate a new set of shared session keys for local secure link again.

But if the length of message $M(x)$ is shorter than $n$, $n$ is the length of $V(v)$ and the maximal length of the data message. Then we will fill the message, and let the length as $M$. In this way, we can use the above scheme to encrypt the message $M(x)$.

## 5 Scheme analysis

We first analytically compare the performance of the proposed session key management scheme with previously proposed group rekeying schemes, then, we conduct simulations to show our scheme have significantly improve the performance in terms of power saving, communication cost, computational overhead, security analysis.

At first, before analyzing the performance of different schemes, we define some notations that are used in this section as follows:

$N$: The total number of the whole nodes in WSNs.

$n$: The average neighbor nodes that a node has.

$L$: The length of the group key. This is the same length of Message data.

### 5.1 Energy Efficient

Most of the power consumed in WSNs is to send and receive message. If we can reduce the traffic of WSNs, we can prolong the lifetime of battery in sensor nodes. In our scheme, we don't need any special phase for update the shard session keys, and we only need to send a random number after each of the shared keys is used. We reduce much traffic in WSNs. Comparing to group rekeying scheme. For example, in time period $[T_0, T_1]$,the average number message of each node is $m_0$ ,and rekeying period is $T'$, so use rekeying schemes, each node has send and received total number

of message is $(n+1)[m_0 + \frac{T_1 - T_0}{T'}]$ in period $[T_0, T_1]$, from above, we know the exchanged message has relate to the number of neighbor $n$ and the rekeying period, $T'$ in many case, considering the security, $T'$ is very short, then the number of message become lots. But using our scheme, we only need to send $m_0$ number message each node in period $[T_0, T_1]$.The result shown as Figure 4 verify that this scheme is quite efficient

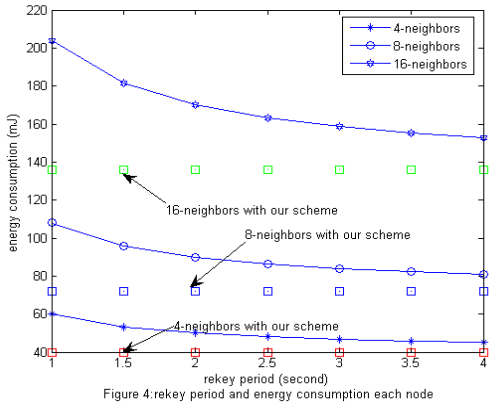

Figure 4:rekey period and energy consumption each node

### 5.2 Communication Cost

In our scheme, each innocent nodes needs to send out $n$ messages to its trusted neighbors when the node has the message to send, each message has $L$ bits. Therefore, $n \bullet L$ bits are sent out by the node. After the message bas been received. The node only need to load the next shared keys form its memory.

### 5.3 security analysis

In this section, we will discuss about the security of our scheme. Form our scheme, each message has a different shared session keys to encrypt the message. Even the same message sent in consecutive twice, we still can get the different cryptograph. In theory, our shared session keys is unbreakable with our realistic attack model.

## 6. Conclusion

We have presented a lightweight session key management scheme for WSNs, which based on a random number to generate a set of shared session keys. Our scheme has a number of appealing properties. First, it is energy efficient. Nodes don't need special phase to update the shared group keys. Second, compared to existing group rekey schemes, it is more resilient against node capture. Furthermore, we have shown that even the same message, $M(x)$ to be encrypted, use our different shared session keys, the vector $Ev(M(x))$ is different every time, so the security can be improved significantly. Third, our scheme needs not much storage requirements and computational overhead in the sensor nodes. Finally, our scheme does not have any rekeying delay applying to a large scale networks.

## 7. References


[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[2] D. W. Carman, P. S. Kruus and B. J. Matt, "Constraints and Approaches for Distributed Sensor Network Security," dated September 1, 2000.NAI Labs Technical Report #00-010, available at *http://download.nai.com/products/media/nai/zip/nailabs-report-00-010-final.zip*

[3] G. Jolly, M.C Kuscu., P. Kokate, M Younis., A low-energy key management protocol for wireless sensor networks, Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on , 30 June-3 July 2003 Pages:335 - 340 vol.1.

[4]L. Eschenauer and V. Gligor, A Key-Management Scheme for Distributed Sensor Networks. In Proc. Of ACM CCS November 2002.

[5]H. Chan, A.Perrig, and D.Song, Random key predistribution schemes for sensor networks, Security and Privacy, 2003. Proceedings. 2003 Symposium on , 11-14 May 2003 Pages:197 – 213.

[6] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod Varshney. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS), Washington DC, October 27-31, 2003.

[7] Sencun Zhu, Shouhuai Xu, Sanjeev Setia, and Sushil Jajodia. Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach. In Proc. of the 11th IEEE International Conference on Network Protocols (ICNP'03), Atlanta, Georgia, November 4-7, 2003.

[8] Wenliang Du, Jing Deng, Yunghsiang S. Han, Shigang Chen and Pramod Varshney. A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. In Proceedings of the IEEE INFOCOM'04, March 7-11, 2004, Hongkong. Pages 586-597.

[9] http://world.std.com/~franl/crypto/

[10] G. Wang, W. Zhang, G. Cao, and T. La Porta, "On Supporting Distributed Collaboration in Sensor networks," IEEE Military Communications Conference (MILCOM), October 2003.

[11] H. Hugh, C. Muckenhirn, and T. Rivers, "Group Key Management Protocol Architecture," Request for comments (RFC) 2093, InternetEngineering Task Force, March 1997.

[12] D. Wallner, E. Harder, and R. Agee, "Key Management for Multicast:Issues and Architectures,