Attack-Resilient Random Key Distribution Scheme For Distributed Sensor Networks

Firdous Kausar[1], Sajid Hussain[2], Tai-hoon Kim[3], and Ashraf Masood[1]

[1] College of Signals, NUST, Rawalpindi, Pakistan.
firdous.imam@gmail.com, ashrafm61@gmail.com
[2] Jodrey School of Computer Science, Acadia University, Nova Scotia, Canada.
sajid.hussain@acadiau.ca,
[3] School of Multimedia, Hannam University, Daejeon, Korea.
taihoonn@empal.com

Abstract. Key pre-distribution schemes are a favored solution for establishing secure communication in sensor networks. Often viewed as the safest way to bootstrap trust, the main drawback is seen to be the large storage overhead imposed on resource-constrained devices and also these schemes are quite insecure because pre-loading global secrets onto exposed devices strengthens the incentive for attackers to compromise nodes. To overcome these drawback, we propose a new key pre-distribution scheme for pairwise key setup in sensor networks. In our scheme each sensor node is assigned with small number of randomly selected generation keys instead of storing big number of random keys and a shared secrete key can be efficiently computed from it. After generating the keys with neighbors the initial keys rings are being deleted from nodes memory. The analysis of our approach shows that it improves the previous random key pre-distribution schemes by providing the more resiliency against node capture and collusion attacks. Even if a node being compromised, an adversary can only exploit a small number of keys nearby the compromised node, while other keys in the network remain safe.

1 Introduction

A wireless sensor network typically consists of a potentially large number of incredibly resource constrained sensor nodes. Each sensor node is usually battery powered, and has a low-end processor, a limited amount of memory, and a low power communication module capable of short-range wireless communication. Their lifetime is determined by their ability to conserve power. The sensor nodes form an ad-hoc network through the wireless links. There are many technological hurdles that must be overcome for ad hoc sensor networks to become practical though. All of these constraints require new hardware designs, software applications, and network architectures that maximize the motes capabilities while keeping them inexpensive to deploy and maintain. Wireless sensor networks are ideal candidates for a wide range of applications, such as target tracking and monitoring of critical infrastructures[1].

Secret communication is an important requirement in many sensor network applications, so shared secret keys are used between communicating nodes to encrypt data. Some of the major constraints like ad hoc nature, intermittent connectivity, and resource limitations of the sensor networks prevent traditional key management and distribution schemes to be applicable to WSN.

A typical WSN may contain from hundreds to thousands of sensor nodes. So any protocol used for key management and distribution should be adaptable to such scales. Sensor nodes in a WSN possess a unique communication pattern. Therefore, security protocols and most important the key management should take care of these patterns. A Berkeley Miac2 motes has a tiny Atmega Microprocessor and 128 KBytes of programmable flash memory. Hence, running computationally intensive cryptographic algorithms over such tiny embedded system devices is infeasible. Public key cryptography is therefore almost ruled out for serving security in WSNs. To avoid the use of public key cryptography, several alternative approaches have been developed to perform key management on resource-constrained sensor networks, such as random key pre-distribution schemes, plain text key exchange schemes, and transitory master key schemes.

Rest of paper is organized as follows. Section 2 provides the related work and Section 3 describes the Threat Model. Section 4 give the problem statement. In Section 5 proposed scheme is described. Section 6 gives the results and performance evaluation. Finally, Section 7 concludes the paper.

2 Related Work

The key management problem in wireless sensor networks has been studied in regard to different objectives and metrics. Eschenauer and Gligor [2] propose a distributed key establishment mechanism that relies on probabilistic key sharing among the nodes of a random graph and uses a shared-key discovery protocol for key establishment. Chan et al. further extend this idea and propose the q-composite key predistribution [3]. This approach allows two sensors to setup a pairwise key only when they share at least q common keys. Chan et al. also develop a random pairwise keys scheme to defeat node capture attacks. Leonardo B. Oliveira et al's [4] show how random key predistribution, widely studied in the context of flat networks, can be used to secure communication in hierarchical (cluster-based) protocols such as LEACH [5]. They present SecLEACH, a protocol for securing node-to-node communication in LEACH-based networks. These and some others [6],[7],[8],[9],[10] efforts have assumed a deployment of homogeneous nodes, and have therefore suggested a balanced distribution of random keys to each of the nodes to achieve security. Most of these schemes are suffered from high communication and computation overhead, and/or high storage requirement.

In [11] Perrig et al. propose SPINS, a security architecture specifically designed for sensor networks. In SPINS, each sensor node shares a secret key with the base station. Two sensor nodes cannot directly establish a secret key. However, they can use the base station as a trusted third party to set up the secret key.

Blundo et al.[12] propose several schemes which allow any group of t parties to compute a common key while being secure against collusion between some of them. These schemes focus on saving communication costs while memory constraints are not placed on group members. When t = 2, one of these schemes is actually a special case of Bloms scheme[13].

Availability of some information on the sensor distribution in the field helps to improve the security of the key pre-distribution schemes. Some location-aware schemes are propose in [14] and [15]. These techniques divide the target field into non-overlapping square areas and randomly deploy the sensors in every area. The exact location of a sensor in any area is unknown, but there is knowledge about the identity of sensors in every area. This information helps to eliminate the dependency of keys between nonadjacent cells.

Du et al. [16] combine the key predistribution scheme of Blom[13] with the random key predistribution of [2] and propose a new scheme using multiple key spaces in which they first construct ω spaces using Blom's scheme, and then have each sensor node carry key information from τ (with $2 \leq \tau < \omega$) randomly selected key spaces. Two nodes can compute a shared key with high probability if they carry key information from a common space.

3 Threat Model

Sensor networks are often deployed in hostile environments, yet nodes cannot afford expensive tamper-resistant hardware. Therefore, a motivated attacker can compromise (via physical or remote exploitation) a set of nodes, obtaining their pairwise secret keys and controlling outbound communications. We also assume nodes can collude by sharing their keys with other attacker nodes. Traditionally, the threat from node compromise is measured by its impact on confidentiality, whether secret keys shared between uncompromised nodes can be obtained.

4 Problem Statement

In random key pre-distribution(RKP) schemes, a large pool of random symmetric keys and their ids is generated, and then every node is assigned with a number of keys randomly selected from a pool. After deployment, nodes broadcast ids of keys along with their node id to neighbors to determine their shared pairwise keys. If the network density, the size of the key pool, and the number of keys assigned to each sensor node are carefully chosen,then it can be ensured with high probability that all the neighboring nodes in the network will share at least one key with each other.While pre-distributing pairwise keys does protect confidentiality, it still loads nodes with a large number of globally-applicable secrets. By eliminating the eavesdropping attack, the pairwise scheme makes another type of malicious behavior more attractive. As several nodes possess the same keys; any node can make use of them. Simply combining the keys obtained

Notation	**Definition**
$f(m,k)$	Pseudorandom function applying on message m using key k
$H(k,m)$	One-way hash function applying on message m using key k
id_a	Identity of node N_a
R_a	Set of the keys in node N_a initial key ring
$\acute{R_a}$	Set of the keys in node N_a updated key ring
k_i^a	i-th key in node N_a key ring
k_i^{na}	i-th key in node N_a updated key ring
$K_{x,y}$	A shared key between node N_x and N_y
$\|\|$	concatenation symbol

Table 1. Symbol Definition

from a significant number of compromised nodes greatly increases the attacker's chances of sharing keys with other nodes. A collusive attacker can share its pairwise keys between compromised nodes, enabling each to present multiple 'authenticated' identities to neighboring nodes while escaping detection [17]. In order to countering the collusion attacks, nodes should destroy unused keys from the node memory after an initialization phase, but this means new nodes can no longer join the system once initialization is complete.

5 Proposed Scheme

In this section we describe our key management scheme in detail. Table 1 shows the notation to be used latter.

Generate a key pool P consist of S different random keys which are called generation keys and their ids prior to network deployment. Shared pairwise keys are generated independently via these generation keys by applying a keyed hash algorithm on it. Before deploying the nodes, each node is loaded with its assigned key ring R which consist of m number of generation keys which are used as the generation knowledge of a number of keys.

For each node N_x, the assigning rules are as follows [18].For every key $k_i \in P$ where $P = (k_1, k_2, ..., k_S)$, compute $z = f(id_x, k_i)$; then, put k_i into R_x, the key ring of node N_x, if and only if $z \equiv\ 0\ mod\ (\frac{S}{m})$. This way we will fills R_x with m keys.

In the shared key discovery phase each node discovers its neighbor in wireless communication range with which it shares keys. The algorithm shown in Figure 1 will be executed on each node during shared key discovery phase. Each node broadcast its id to the neighboring nodes. The neighboring nodes which receive the message, compute the set of their key ids in order to find shared keys as follows. Consider a node N_a that is willing to know which keys it shares with its neighbors. It broadcast its id and wait to receive same broadcast message from neighboring nodes. Suppose it receive message from node N_b, it extract the node id from message i.e. id_b. For every key $k_j^a \in R_a$ node N_a computes $z = f(id_b, k_j^a)$

. If $z \equiv\ 0\ mod(\frac{S}{m})$, it means that node N$_b$ also has key k$_j^a$ in its key ring i.e. $R_a \cap R_b = k_j^a$.

They will generate the shared pairwise key by applying keyed hash algorithm on id_a and id_b by using k_j^a , K$_{a,b}$ $= H(k_j^a, id_a||id_b)$.After generating the shared keys with neighbors, each node destroy its initial key ring.

```
/* Initial State */
$S$: Key Pool
$R$: Key Ring
$m$: number of keys in key pool

procedure SharedKeyDiscovery()
broadcast(id$_a$)
while receive=TRUE do
    packet=recieveBroadcastMsgs();
    id$_b$=packet.getNodeId();
    for $\forall k_j^a \in R_a$ do
        $z = f(id_b, k_j^a)$
        if ($z \equiv\ 0\ mod(\frac{S}{m})$) then
            $K_{a,b} = H(k_j^a, id_a||id_b)$
        end if
    end for
end while
```

Fig. 1. Shared key discovery algorithm

5.1 Addition after initial deployment

Our approach should support the ability to allow new nodes to join network even after the nodes already present in the network has destroyed their initial key rings. The RKP scheme will be unable to add new nodes once the initial key rings has been deleted from node's memory. As a result, it is imperative that we develop a new solution capable of handling joins beyond the initial deployment. Suppose a newly joining node N_y wants to setup a pairwise key with an existing node N_x. There is a problem that how can N_x who no longer has its initial key ring can come to know that whether it shared any of its key from deleted key ring with N_y?; We propose a solution that addresses this problems and allows new legitimate nodes to join an existing sensor network, while preserving opaqueness before and after erasure of node's key ring.

First, before a node N_x destroys its initial key ring, it generates a new key ring as shown in Figure 2. For every key k_i^x in its key ring, it generate a new key k_i^{nx} by applying pseudorandom function on its id and k_i^x . In this way it generate a set of new keys from keys in its initial key ring and assigned these newly generated keys the same id as that was of original keys in order to keep

```
procedure deleteKeyRing()
for $\forall k_j^x \in R_x$ do
    $k_j^{nx} = f(id_x, k_j^x)$
    $id_{k_j^{nx}} = id_{k_j^x}$
    delete($k_j^x$)
end for
```

Fig. 2. Initial key ring update algorithm

record that which keys it have in its initial key ring. Every new node join the network after initial deployment execute the algorithm shown in Figure 3 in order to discover the shared keys with neighbors.

```
procedure nodeAddition()
broadcast(id$_y$)
while receive=TRUE do
    packet=recieveBroadcastMsgs();
    id$_x$=packet.getNodeId();
    for $\forall k_j^y \in R_y$ do
        $z = f(id_x, k_j^y)$
        if ($z \equiv\ 0\ mod(\frac{S}{m})$ ) then
            $k_j^{nx} = f(id_x, k_j^y)$
            $K_{x,y} = H(k_j^{nx}, id_x || id_y)$
        end if
    end for
end while
```

Fig. 3. New node addition algorithm

Suppose new node N_y wants to join a network, it broadcast its node id (id_y) and wait for reply. When it receive reply message (say from node N_x), it extract node id. For every key k$_j^y$ $\in R_y$ node N_y computes $z = f(id_x, k_j^y)$. If $z \equiv\ 0\ mod(\frac{S}{m})$, it means that node N_x also has key k_j^y in it initial key ring but it is no longer available now. So N_y computes corresponding key i.e. k_j^{nx} of N_x new key ring by applying pseudorandom function on id_x and k_j^y. Now they will generate the shared pairwise key by applying keyed hash algorithm on id_a and id_b by using k_j^{nx} i.e. $K_{x,y} = H(k_j^{nx}, id_x || id_y)$.

6 Analysis

Earlier section described various steps in the proposed key management scheme. This section analyzes the algorithm as a whole to explain its features that make this scheme feasible to implement and better alternative option, compared to other similar key management algorithms.

To make it possible for any pair of nodes to be able to find a secret key between them, the key sharing graph G$_{ks}$(V,E) needs to be connected. Given the size and the density of a network, how can we select the values for S and m , s.t., the graph G$_{ks}$ is connected with high probability? We use the following approach, which is adapted from [2].

Let P_c be the probability that the key-sharing graph is connected. We call it global connectivity. We use local connectivity to refer to the probability of two neighboring nodes find at least one common key in their key rings. The global connectivity and the local connectivity are related: to achieve a desired global connectivity P_c, the local connectivity must be higher than a certain value; we call this value the required local connectivity, denoted by p$_{required}$.

Using connectivity theory in a random-graph by Erdos and Renyi, we can obtain the necessary expected node degree d (i.e., the average number of edges connected to each node) for a network of size n when n is large in order to achieve a given global connectivity, P_c:

$$d = \frac{(n-1)}{n}[ln(n) - ln(-ln(P_c))] \tag{1}$$

For a given density of sensor network deployment, let $\acute{n}$ be the expected number of neighbors within wireless communication range of a node. Since the expected node degree must be at least d as calculated above, the required local connectivity p$_{required}$ can be estimated as:

$$p_{required} = \frac{d}{\acute{n}} \tag{2}$$

After we have selected values for S and m , the actual local connectivity is determined by these values. Let p$_s$ is the probability of any two neighboring nodes sharing at least one common key in their key rings

$$p_s = 1 - \acute{p}_s \tag{3}$$

Where p$_{\overline{s}}$,the probability that they will not share a key, is given by:

$$\acute{p}_s = \frac{[(S-m)!]^2}{S!(S-2m)!} \tag{4}$$

Knowing the required local connectivity p$_{required}$ and the actual local connectivity p$_S$, in order to achieve the desired global connectivity P_c, we should have p_s≥ p$_{required}$.

$$1 - \frac{[(S-m)!]^2}{S!(S-2m)!} \geq \frac{(n-1)}{n\acute{n}}[ln(n) - ln(-ln(Pc))] \tag{5}$$

Therefore, in order to achieve a certain P_c for a network of size n and the expected number of neighbors for each node being $\acute{n}$, we just need to find values of S and m, such that Inequality (5) is satisfied.

We simulated a sensor network comprised of nodes uniformly distributed over a plane, setting n = 1000, $\acute{n} = 40$,and P_c=0.99. In this setting the value of $P_{required}$ is 0.25. so we have to find the values of S and m such that $ps \geq 0.25$. In Figure 4 we show the probability of key sharing i.e. ps among nodes for different values of S and m.

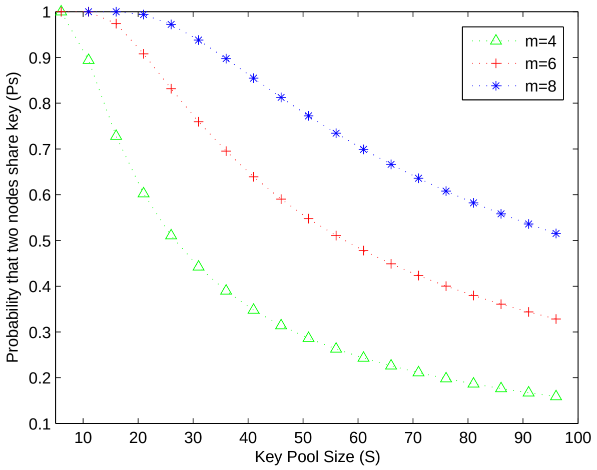

Fig. 4. The Key Sharing Probability

6.1 Security Analysis

We evaluate our key pre-distribution scheme in terms of its resilience against node capture and collusion attack.

Our evaluation is based on the metric that how much of network communication an adversary can compromise, when x nodes are captured. To compute this fraction, we first compute the probability that any one of the additional communication links is compromised after x nodes are captured. In our analysis, we are considering the links which are secured using a pairwise key computed from the common generation key shared by the two nodes of this link.

We should also notice that during shared key discovery process, two neighboring nodes find the common generation key in their key rings and use this key to agree upon another random key to secure their communication. Because this new key is generated from generation key by applying keyed hash algorithm on it, the security of this new random key does not directly depend on whether the key rings are broken. However, if an adversary can record all the communications during the key setup stage, he/she can still compromise this new key after compromising the corresponding links in the network. The fraction of communications compromised when x number of nodes being compromised is shown in Figure 5 in which we give the comparison of our proposed scheme (PS) with basic scheme (EG) and q-composite scheme.

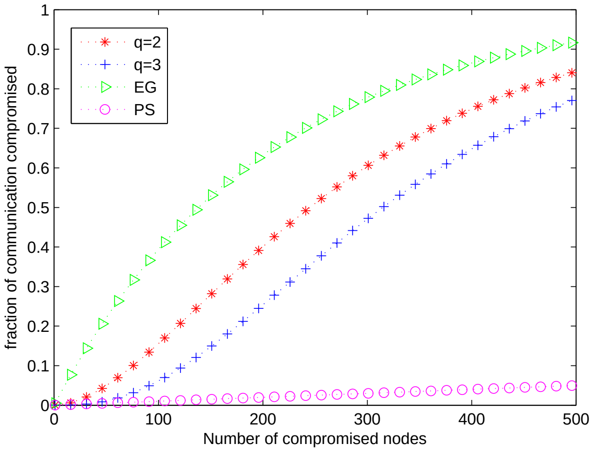

Fig. 5. The Compromising Probability

Colluding attackers mainly take advantage of the pairwise secret keys stored by each sensor node as these keys are globally-applicable secrets and can be used throughout the network, yet ordinary sensors can only communicate with the small fraction of nodes within radio range. An attacker can readily exploit this lack of coordination between nodes and can now share its pairwise keys between compromised nodes, enabling each to present multiple 'authenticated' identities to neighboring nodes while escaping detection.In our proposed scheme we deleted the initial key ring from nodes memory after setting up shared pairwise keys with neighbors. Instead nodes generate a new key ring locally from initial key ring by applying one way hash function on node id and keys in its key ring. So no more globally-applicable secretes remains in the node memory and It is not possible by adversary to launch this attack.

7 Conclusion

In this paper we proposed a key distribution scheme for wireless sensor networks which is secure against collusion attack. The analysis shows that the proposed scheme provide more resiliency against node capture and collusion attack by deleting the initial key rings from their memory after generating the shared pairwise key with neighbors. It also allow new nodes to join the system once initialization is complete and initial key ring has been destroyed from node's memory.

Acknowledgment

This work is in part supported by Higher Education Commission (HEC) Pakistan's International Research Support Initiative Program's scholarship given to Firdous Kausar to conduct her research at Acadia University, Canada, under supervision of Dr. Sajid Hussain. Further, the work was partly supported by Dr. Hussain's National Science and Engineering Research Council (NSERC) Canada's RTI and Discovery grants.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine (2002)
2. Eschenauer, L., Gligor, V.D.: A key management scheme for distributed sensor networks. In: ACM CCS. (2002)
3. Chan, H., Perrig, A., Song, D.: Random key pre-distribution schemes for sensor networks. In: IEEE Symposium on Security and Privacy. (2003) 197–213
4. Oliveira, L.B., Wong, H.C., Bern, M., Dahab, R., Loureiro, A.A.F.: Sec leach: A random key distribution solution for securing clustered sensor networks. In: 5th IEEE international symposium on network computing and applications. (2006) 145–154
5. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: IEEE Hawaii Int. Conf. on System Sciences. (2000) 4–7
6. Chan, H., Perrig, A., Song, D.: Random key pre-distribution schemes for sensor networkss. In: IEEE Symposium on Research in Security and Privacy. (2003)
7. Zhu, S., Xu, S., Setia, S., Jajodia, S.: Establishing pairwise keys for secure communication in ad hoc networks: A probabilistic approach. In: 11th IEEE International Conference on Network Protocols (ICNP'03). (2003)
8. Pietro, R.D., Mancini, L.V., Mei, A.: Random key assignment secure wireless sensor networks. In: 1st ACM workshop on Security of Ad Hoc and Sensor Networks. (2003)
9. Cheng, Y., Agrawal, D.P.: Efficient pairwise key establishment and management in static wireless sensor networks. In: Second IEEE International Conference on Mobile ad hoc and Sensor Systems. (2005)
10. Ren, K., Zeng, K., Lou, W.: A new approach for random key pre-distribution in large-scale wireless sensor networks. Wireless communication and mobile computing **6**(3) (2006) 307–318
11. Perrig, A., Szewczyk, R., Tygar, J., Victorwen, Culler, D.E.: Spins: Security protocols for sensor networks. In: Seventh Annual Int'l Conf. on Mobile Computing and Networks. (2001)
12. Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, London, UK, Springer-Verlag (1993) 471–486

13. Blom, R.: An optimal class of symmetric key generation systems. In: Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques, New York, NY, USA, Springer-Verlag New York, Inc. (1985) 335–338
14. Liu, D., Ning, P.: Location-based pairwise key establishments for static sensor networks. In: SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, New York, NY, USA, ACM Press (2003) 72–82
15. Wadaa, A., Olariu, S., Wilson, L., Eltoweissy, M.: Scalable cryptographic key management in wireless sensor networks. In: ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04), Washington, DC, USA, IEEE Computer Society (2004) 796–802
16. Du, W., Deng, J., Han, Y.S., Varshney, P.K., Katz, J., Khalili, A.: A pairwise key predistribution scheme for wireless sensor networks. ACM Trans. Inf. Syst. Secur. **8**(2) (2005) 228–258
17. Moore, T.: A collusion attack on pairwise key predistribution schemes for distributed sensor networks. In: PERCOMW '06: Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops, Washington, DC, USA, IEEE Computer Society (2006) 251
18. Pietro, R.D., Mancini, L.V., Mei, A.: Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks. Wirel. Netw. **12**(6) (2006) 709–721