

Term Report for Course: 4005-898-01 Independent Study, August 2008

**IMPLEMENTATION OF
RANDOM PAIR-WISE KEY PRE-DISTRIBUTION SCHEME
ON SUN SPOT WIRELESS SENSORS**

*Wisam F. Kadhim
Dept. of Computer Science
Rochester Institute of Technology
wfk9421@cs.rit.edu*

Abstract

Providing means of secure communication in Wireless Sensor Networks depends on proper session keys exchange, where typical Diffie-Hellman Key Exchange and Public Key Cryptography are inapplicable on Wireless Sensors due to their limited resources. Random Pair-wise Key Pre-Distribution is one of the recently developed schemes to provide secure means of exchanging pair-wise session keys that are generated probabilistically. In this report we present background analysis on probabilistic key pre-distribution schemes, as well as implementation details of Random Pair-wise Key Pre-distribution scheme on Sun SPOT wireless sensors.

Contents

Abstract	<i>Page 1</i>
I. Introduction	<i>Page 3</i>
II. Background on Random Key Pre-Distribution Schemes	<i>Page 4</i>
III. Random Pair-wise Key Scheme	<i>Page 7</i>
III.1 Description of the random pair-wise scheme	<i>Page 8</i>
III.2 Initialization and key-setup in random pair-wise keys scheme	<i>Page 8</i>
III.3 Multi-hop range extension	<i>Page 9</i>
IV. Implementation	<i>Page 10</i>
IV.1 Pre-Deployment (Initialization) Phase	<i>Page 11</i>
IV.2 Post-Deployment (Key-Setup) Phase	<i>Page 11</i>
V. Conclusions	<i>Page 13</i>
References	<i>Page 14</i>

I. Introduction

Wireless Sensor Networks (WSN) are networks of small, battery-powered, memory-constraint devices named sensor nodes, which have the capability of wireless communication over a restricted area. Due to memory and power constraints, they need to be well arranged to build a fully functional network.

Security in WSN has several challenges: wireless nature of communication, resource limitation on sensor nodes, very large and dense WSN, lack of fixed infrastructure, unknown network topology prior to deployment and high risk of physical attacks to unattended sensors. Moreover, in some deployment scenarios sensor nodes need to operate under adversarial conditions. Security solutions for such applications depend on existence of strong and efficient key distribution mechanisms. It is infeasible, or even impossible in uncontrolled environments, to visit a large number of sensor nodes, and change their configuration. Also, use of a single shared key in whole WSN is not a good idea because an adversary can easily obtain the key. Thus, sensor nodes have to adapt to their environments and establish a secure network by: using pre-distributed keys or keying materials, exchanging information with their immediate neighbors, or exchanging information with computationally robust nodes [10].

Key pre-distribution is a method of distributing keys onto nodes before deployment. Therefore, the nodes build up the network using their secret keys after deployment, that is, when they reach their target position.

Key pre-distribution schemes are various methods that have been developed by academicians for a better maintenance of key management in WSNs. Basically a key pre-distribution scheme has three phases, key distribution, shared key discovery and path-key establishment. During these phases, secret keys are generated, placed in sensor nodes, and each sensor node searches the area in its communication range to find another node to communicate. A secure link is established when two nodes discover one or more common keys (this differs in each scheme), and communication is done on that link between those two nodes. Afterwards, paths are established connecting these links, to create a connected graph. The result is a wireless communication network functioning in its own way, according to the key pre-distribution scheme used in creation [6].

In section II of this paper, we present a background analysis on the main scheme developed by Eschenauer and Gligor [3] that introduced the idea of random key distribution, and then discuss the several enhancements over the main scheme. Section III gives details of random pair-wise key pre-distribution scheme developed by Chan, Perrig and Song [2] and its enhancements over the main scheme. The rest of the paper shows the implementation details of random pair-wise key pre-distribution scheme on Sun SPOT wireless sensors and conclusions.

II. Background on Random Key Pre-distribution Schemes

In 2002 Echenauer and Gligor proposed the idea of basic probabilistic key pre-distribution scheme, which relies on probabilistic key sharing among the nodes of a random graph [3]. In key setup phase, a large key-pool of N keys and their identities are generated. For each sensor, m keys are randomly drawn from the key-pool N without replacement. These m keys and their identities form the key-chain for a sensor node. Thus, probability of key share among two sensor nodes becomes:

$$P = \frac{((N-m)!)^2}{((N-2m)!N!)}.$$

In shared-key discovery phase, two neighbor nodes exchange and compare list of identities of keys in their key-chains. Basically, each sensor node broadcasts one message, and receives one message from each node within its radio range where messages carry key ID list of size m [10].

Cluster key grouping scheme developed by Hwang [4] proposes to divide key-chains into C clusters where each cluster has a start key ID. Remaining key IDs within the cluster are implicitly known from the start key ID. Thus, only start key IDs for clusters are broadcasted during shared-key discovery phase which means messages carry key ID list of size c instead of m [10]. Another scheme is given by *Pair-wise key establishment protocol* developed by Zhu [11] which requires every sensor node to have a unique ID which is used as a seed to a Pseudo Random Function (PRF). Key IDs for the keys in the key-chain of node S_A are generated by $\text{PRF}(\text{ID}_A)$. Thus, broadcast messages carry only one key ID. Also, storage, which is required to buffer received broadcast message before processing, decreases substantially. But, a sensor node has to execute $\text{PRF}(\text{ID})$ for each broadcast message received from a neighbor. *Transmission range adjustment scheme* developed by Hwang and Kim [5] proposes sensor nodes to increase their transmission ranges during shared-key discovery phase. Nodes return to their original optimal transmission range once the keys are discovered. Idea is to decrease communication burden in path-key establishment phase, and to save energy while still providing a good key connectivity [10].

It is possible to protect key identities broadcasted in shared-key discovery by using a method similar to Merkle Puzzle [8] which substantially increases processing and communication usage. After shared-key discovery phase, some node pairs may not be able to find a key in common. These pairs apply path-key establishment phase to communicate securely through other nodes. Scalability and resilience of the solutions can be improved by using larger key pools. But, larger key-pool means smaller probability of key share because key-chain size may not increase due to storage limitations. Probability that a link is compromised, when a sensor node is captured, is m/N which is very high for small key-pools, and produces low resilience [10].

There are several key reinforcement proposals to strengthen security of the established link keys, and improve resilience. Objective is to securely generate a unique link or path-key by using established keys, so that the key is not compromised when one or more sensor node is captured. One approach is to increase amount of key overlap required in shared-key discovery phase. *Q-composite random key pre-distribution scheme* developed by Chan [2] requires q common keys to establish a

link key. Link key $K_{A,B}$ between a pair of sensor nodes S_A and S_B is set as hash of all common keys $K_{A,B} = \text{Hash}(K_1 || K_2 || K_3 || \dots || K_q)$. The scheme improves resilience because probability that a link is compromised, when a sensor node is captured, decreases from m/N to $\binom{m}{p} / \binom{N}{p}$. But, probability of key sharing also decreases because a pair of nodes has to share q keys instead of one [10].

Another approach is to reinforce the established link key. In *Multi-path key reinforcement scheme* developed by Chan [2], node S_A generates j random key updates rk_i and sends them through j disjoint secure paths. S_B can generate reinforced link key:

$$K_{A,B}^r = K_{A,B} + rk_1 + \dots + rk_j$$

upon receiving all key updates. This approach requires nodes S_A and S_B to send and receive j more messages each of which carries a key update. Moreover, each node on the j disjoint path has to send and receive an extra message. Similar mechanism is proposed by *Pair-wise key establishment protocol* developed by Zhu [11] which uses threshold secret sharing for key reinforcement. S_A generates a secret key $K_{A,B}^r$, $j - 1$ random shares sk_i , and:

$$sk_i = K_{A,B}^r + sk_1 + \dots + sk_{j-1}$$

S_A sends the shares through j disjoint secure paths. S_B can recover $K_{A,B}^r$ upon receiving all shares [10].

In Co-operative pair-wise key establishment protocol developed by Pietro [9], S_A first chooses a set $C = \{c_1, c_2, \dots, c_m\}$ of co-operative nodes. A co-operative node provides a hash $\text{HMAC}(K_{c1,B}, \text{ID}_A)$. Reinforced key is then:

$$K_{A,B}^r = K_{A,B} + \sum_{c \in C} \text{HMAC}(K_{c,B}, \text{ID}_A)$$

where $K_{A,B}$ and $K_{c,B}$ are the established link keys. Node S_A shares set C with node S_B , therefore, S_B can generate the same key. This approach requires nodes S_A and S_B to send and receive c more messages. Moreover, cooperative nodes have to send and receive two extra messages. In addition to increased communication cost, each cooperative node has to execute HMAC function twice for S_A and S_B . The key reinforcement scheme in general increase processing and communication complexity, but provide good resilience in the sense that a compromised key-chain does not directly affect security of any links in the WSN. But, it may be possible for an adversary to recover initial link keys. An adversary can then recover reinforced link keys from the recorded multi-path reinforcement messages when the link keys are compromised [10].

Sensor nodes, which are far away from each other, do not need to have common keys in their key-chains. Similar to *Closest pair-wise keys pre-distribution scheme* [7], *Key pre-distribution by using deployment knowledge scheme* developed by Du [1] uses location information. It models deployment knowledge and develops a key pre-distribution scheme based on the model. The scheme divides sensor nodes into $t \times n$ groups $G_{i,j}$ and deploys them at a resident point (x_i, y_j) for $1 \leq i \leq t$ and $1 \leq j \leq n$ where the points are arranged as two dimensional grids. Resident points of a node $m \in G_{i,j}$ follows the pdf:

$$f_m^{i,j}(x, y | m \in G_{i,j}) = f(x - x_i, y - y_j)$$

where $f(x, y)$ is a two dimensional Gaussian distribution. In key setup phase, key-pool N is divided into $t \times n$ key-pools $N_{i,j}$ of size $\omega_{i,j}$. The pool $N_{i,j}$ is used as key-pool for the nodes in group $G_{i,j}$. Given $\omega_{i,j}$ and overlapping factors α and β , key-pool is divided into subsets where: (i) two horizontally and vertically neighboring key-pools have $\alpha \times \omega_{i,j}$ keys in common, (ii) two diagonally neighboring key-pools have $\beta \times \omega_{i,j}$ keys in common, and (iii) non-neighboring key-pools do not share a key. Basic probabilistic key pre-distribution scheme is applied within each group. Problem in this scheme is the difficulty to decide on parameters $\omega_{i,j}$, α and β to provide a good key connectivity [10].

III. Random Pair-wise Key Scheme

In sensor network security, an important challenge is the design of protocols to bootstrap the establishment of a secure communications infrastructure from a collection of sensor nodes which may have been pre-initialized with some secret information but have had no prior direct contact with each other. This problem is referred to as the *bootstrapping problem*. A bootstrapping protocol must not only enable a newly deployed sensor network to initiate a secure infrastructure, but it must also allow nodes deployed at a later time to join the network securely. The difficulty of the bootstrapping problem stems from the numerous limitations of sensor networks, some of the more important ones include the inability to utilize existing public key cryptosystems (since the expensive computations involved could expose the power-constrained nodes to a denial-of-service attack), the inability to pre-determine which nodes will be neighbors after deployment, and the inability of any node to put absolute trust in its neighbor (since the nodes are not tamper resistant and are vulnerable to physical capture) [2].

In the random key schemes presented in previous section, while each node can verify that some of its neighbors have certain secret keys and are thus legitimate nodes, no node can authenticate the identity of a neighbor that it is communicating with. For example, suppose node S_A shares some set of keys K with node S_B and that they use these keys as the basis for securing a communications link. Because keys can be issued multiple times out of the key pool, other nodes, e.g., S_C could also hold this set of secret keys K in its key ring. S_A cannot ascertain that it is really communicating with S_B and not S_C , since it knows nothing more about S_B than its knowledge of K [2].

In 2003, Chan, Perrig, and Song proposed a new key establishment protocol called the *random pair-wise scheme* that possesses the key property of node-to-node authentication. The random pair-wise scheme has the following properties:

- *Perfect resilience against node capture.* Any node that is captured reveals no information about links that it is not directly involved in.
- *Node-to-node identity authentication.* Nodes are able to verify the identities of the nodes with whom they are communicating. An adversary is unable to impersonate the identity of any node S_B unless S_B has already been captured.
- *Distributed Node Revocation.* With some added overhead in key storage, misbehaving nodes can be revoked from the network without involving a base station.
- *Resistance to node replication and generation.* The scheme can reduce the opportunity of node replication at some cost to node memory and communication setup overhead.
- *Comparable scalability.* The scheme can support a maximum number of nodes that is comparable to the number of nodes supportable by the basic scheme and q-composite schemes under the limited global payoff requirement [2].

III.1 Description of the random pair-wise scheme:

Suppose a sensor network has a maximum of N nodes. A simple solution to the key pre-distribution problem is the *pair-wise* keys scheme where each node contains $N-1$ communication keys each being pair-wise privately shared with one other node in the network [2].

The *random pair-wise* keys scheme is a modification of the pair-wise keys scheme based on the observation that not all $N-1$ keys need to be stored in the node's key ring to have a connected random graph with high probability. Erdős and Rényi's formula allows us to calculate the smallest probability p of any two nodes being connected such that the entire graph is connected with high probability c . To achieve this probability p in a network with N nodes, each node need only store a random set of Np pair-wise keys instead of exhaustively storing all $N-1$. Reversing the calculation, if a node can store m keys, then the maximum supportable network size is:

$$N = \frac{m}{p}$$

Depending on the model of connectivity, p may grow slowly with N when N is large (intuitively, p cannot decrease as N goes toward infinity since it is more likely that a large graph is disconnected than a smaller graph). Hence, N should increase with increasing m and decreasing p . The exact rates will depend on the deployment model [2].

The use of pair-wise keys instead of purely random keys chosen from a given pool can give us node-to-node authentication properties if each node which holds some key k , also stores the identity (ID) of the other node which also holds k . Hence, if k is used to create a secure link with another node, both nodes are certain of the identity of each other since no other nodes can hold k [2].

III.2 Initialization and key-setup in the random pair-wise keys scheme:

Recall that the size of each node's key rings is m keys, and the probability of any two nodes being able to communicate securely is p . The random pair-wise keys scheme proceeds as follows:

1. In the pre-deployment *initialization* phase, a total of $N = \frac{m}{p}$ unique node identities are generated. The actual size of the network may be smaller than N . Unused node identities will be used if additional nodes are added to the network in the future. Each node identity is matched up with m other randomly selected distinct node IDs and a pair-wise key is generated for each pair of nodes. The key is stored in both nodes' key rings, along with the ID of the other node that also knows the key.
2. In the post-deployment *key-setup* phase, each node first broadcasts its node ID to its immediate neighbors. By searching for each other's IDs in their key rings, the neighboring nodes can tell if they share a common pair-wise key for communication. A cryptographic handshake is then performed between neighbor nodes who wish to mutually verify that they do indeed have knowledge of the key [2].

III.3 Multi-hop range extension:

Since the node ID is just a few bytes, key discovery involves much less network traffic and computational overhead in the nodes than standard random key pre-distribution. Therefore, the effective communication range of nodes for key setup can be extended beyond physical communication range by having neighboring nodes rebroadcast the node ID for a certain number of hops. Each hop that the node ID is rebroadcast effectively extends the range by approximately one communication radius, increasing the number of nodes that can hear the broadcast by a squared factor [2].

This has an impact on the maximum supportable network size N . Based on Main Scheme [3], the connection probability $p = \frac{d}{n'}$, where n' is the number of neighbors and d was computed via the required probability of graph connectivity [2]. From previous equation, we have that maximum network size $N = \frac{m}{p}$ where m is the key ring size. Hence:

$$N = \frac{mn'}{d}$$

By increasing the effective communications radius, we also increase the number of neighbors n' , hence the maximum supportable network size N also increases [2].

Multi-hop range extension should be used with caution, however, because the rebroadcast is performed without verification or authentication. Thus, during the deployment phase, an adversary can send random node IDs into the network which will then be rebroadcast x times by the receiving nodes. This potential denial of service (DoS) attack could stop or slow the key-setup process since the sensor network is actively helping to amplify the range of the adversary's interfering transmissions. The potential damage due to this DoS attack can be reduced by limiting the number of hops of the range extension. If DoS is a serious concern then multi-hop range extension could be removed altogether; it is not required for the operation of the random pair-wise scheme [2].

IV. Implementation

The random pair-wise key pre-distribution scheme (as explained in previous section) consists of two main phases: pre-deployment and post-deployment. Each phase is performed on a different platform, where the pre-deployment (initialization) phase is performed on a resourceful machine (such as a desktop or even a laptop) that can provide enough memory space and processing capacity to generate the pair-wise keys randomly with the expected probability of each two nodes sharing the same key such that the overall network graph is connected with high probability. Yet, the post-deployment (key-setup) phase is performed on resource limited wireless sensors (Sun SPOTS) to broadcast its own generated ID and check for neighbors' response. If it received a neighbor ID broadcast that matches an ID in the ID/key list then it sets up the pair-wise key with that node, otherwise it would be ignored (or re-broadcasted in case of multi-hop mode of operation).

The system architecture consists of two parts (as shown in figure 1): the first part is a random pair-wise key generator to perform the pre-deployment that produces the required number node IDs with their associated pair-wise keys, and upload each sensor with its boot-up information. This boot-up information consists of sensor node ID and list of pair-wise ID/key as generated by the random generator. The second part is a middle layer (between the application and Sun SPOT) that would provide a secure layer for transmission based on the random pair-wise key scheme. This secure layer would perform the post-deployment that sets up pair-wise keys with active neighbors, and use those pair-wise keys to encrypt (and decrypt) data sent (and received) from the application through Sun SPOT to some other peer application on a Sun SPOT sensor that has the same pair-wise key.

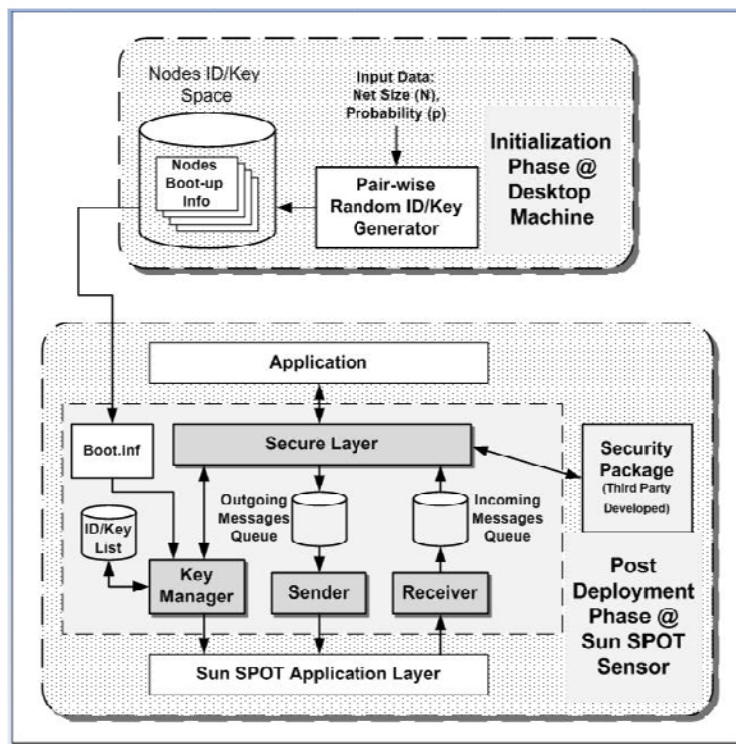


Figure 1: Block Diagram of System Architecture.

IV.1 Pre-Deployment (initialization) phase:

The first task in random pair-wise key generator is generating node IDs at random such that the number of generated IDs equal the size of the designed sensor network N . The next step is cross-link these IDs with each other randomly, such that every node ID is cross-linked with m other IDs according to the probability p . Each cross-linked IDs share a randomly generated key (where this key is unique and obtained only by those two IDs) thus forming the pair-wise key. After the cross-link operation each node ID along with its generated pair-wise keys list are stored in a separate file, where each file would be uploaded to a Sun Spot sensor to provide the required boot-up information for post-deployment phase. An example of pre-deployment (initialization) phase is shown in figure 2 below.

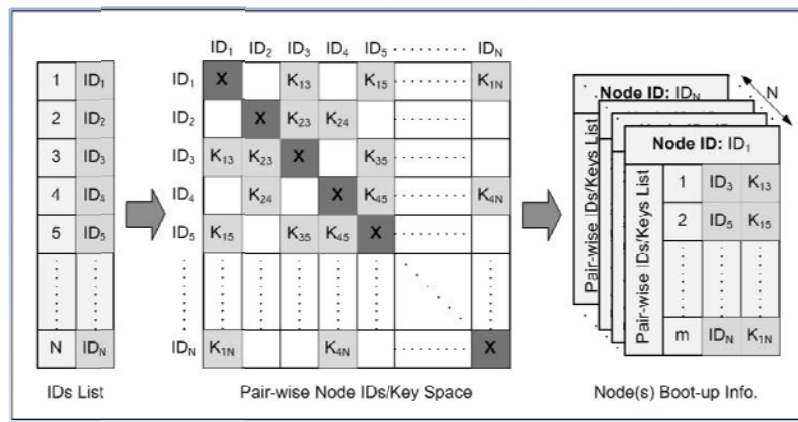


Figure 2: An Example of Pre-Deployment Initialization Phase.

IV.2 Post-Deployment (Key-Setup) Phase:

Every Sun SPOT sensor would be deployed with a *boot.inf* file that has the boot-up information for each sensor and a secure layer package that performs the key-setup phase as well as securing data transmission between the application that runs on Sun SPOT and its peers on other Sun SPOT sensors. The secure layer package consists of 4 classes, each class performs a certain task (as shown in figure 1): Secure Layer class, Key Manager class, Sender class, and Receiver class. The Secure Layer provides a secure communication interface to applications running on top using the pair-wise key management along with encryption and decryption functions.

The secure layer takes the data passed by the application and encrypts it using the pair-wise key that matches the ID of the destination sensor then puts the data on Outgoing Message Queue to be processed by the Sender class. The secure layer also creates a thread that gets received messages from the Incoming Messages Queue and processes them according to the message type, if the message type is *key-setup* then it calls the setup function within the Key Manager class. If the message type is *data* then it notifies the application of message arrival and decrypts the message using the proper pair-wise key to pass it to the application layer.

The Key Manager class controls the key-setup with pair-wise nodes. The first task it performs is sensor boot-up from the boot.inf file to get the sensor ID and constructs the pair-wise key list as a hash table that records the pair-wise IDs, corresponding keys, their address, and activation status. The second task is to verify if the received node ID is included in the pair-wise key list, if it is included, then the ID is marked as active and its node address is recorded so as to be used by secure layer.

The Sender class creates a thread that takes a message stored by the secure layer in the Outgoing Messages Queue and checks the destination address of that message to create a datagram socket and send the message as a datagram to destination sensor. The Receiver class creates a thread to listen to incoming datagram messages to this sensor. Whenever a datagram message is received, the thread stores that message in the Incoming Messages Queue, so the secure layer would fetch messages from this queue and perform the required operation according to the message type.

After each Sun SPOT sensor is deployed, it reads its boot-up information from the file *boot.inf* to get its node ID and the list of pair-wise keys, and then starts the process of key-setup by periodical broadcast of its ID to neighbors within radio transmission range. The delay between broadcasts is set to 5 seconds. During the broadcast process, a receiving thread is instantiated to listen to incoming responses, once there is a response from neighbors the Sun SPOT checks if that response is a key-setup message, and checks if the received ID is included in the pair-wise list. If it is not included, that message is simply ignored, otherwise it marks the ID in the pair-wise list as active, and records the address from which this message was received.

Figure 3 shows an example of a post-deployment scenario, where every sensor broadcasts its' ID and listens to other nodes responses within the radio transmission range. Node S_{ID2} has pair-wise keys with nodes S_{ID1} and S_{ID3} and marked as active since they are within radio range, where as its pair-wise key with node S_{ID7} is not active because it's out of radio range.

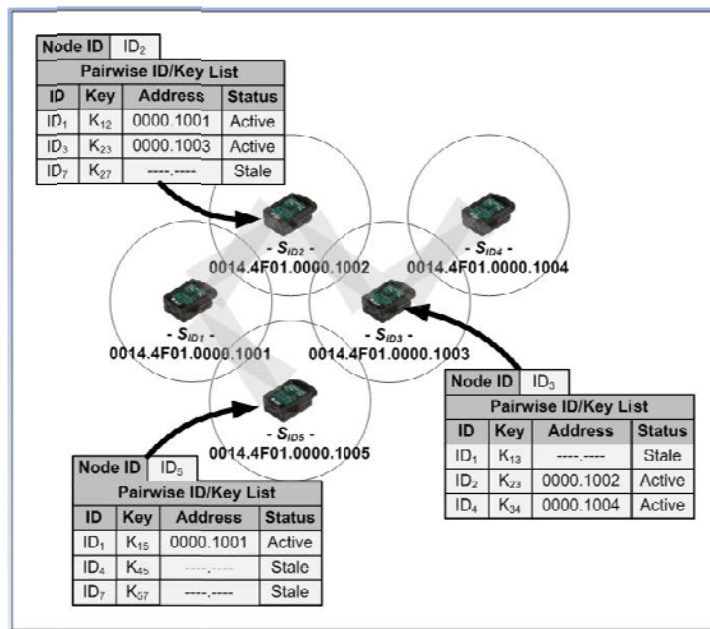


Figure 3: An Example of Post-Deployment Key-Setup Phase.

V. Conclusions

Wireless sensor networks have some specific requirements in addition to the general security requirements, these requirements can be summarize as:

- *Survivability*: ability to provide a minimum level of service in the presence of power loss, failures or attacks.
- *Degradation of security services*: ability to change security level as resource availability changes. These security requirements can be provided by a key distribution mechanism with the requirements given below. These are also used as metrics throughout the paper to evaluate key distribution solutions.
- *Scalability*: ability to support larger networks. Key distribution mechanism must support large networks, and must be flexible against substantial increase in the size of the network even after deployment.
- *Efficiency*: storage, processing, and communication limitations on sensor nodes must be considered.
- *Storage complexity*: amount of memory required to store security credentials.
- *Processing complexity*: amount of processor cycles required to establish a key.
- *Communication complexity*: number of messages exchanged during a key generation process.
- *Key connectivity (probability of key-share)*: probability that two (or more) sensor nodes store the same key or keying material. Enough key connectivity must be provided for a WSN to perform its intended functionality.
- *Resilience*: resistance against node capture. Compromise of security credentials, which are stored on a sensor node or exchanged over radio links, should not reveal information about security of any other links in the WSN. Usually, higher resilience means lower number of compromised links.

In general, resource usage, scalability, key connectivity and resilience are conflicting requirements; therefore, trade-offs among these requirements must be carefully observed [10].

The Random pair-wise key scheme [2] addresses unnecessary storage problem, yet provides very good key resilience as well as support for node based revocation and resistance to node replication. Each sensor uses $2m$ units of memory to store its key-chain. In the shared-key discovery phase, each node broadcasts its ID; therefore, each node sends one message, and receives one message from each node within its radio range. Neighboring nodes can tell if they share a common pair-wise key. This solution has very good key resilience. It is more scalable in the sense that efficient use of memory spaces helps support larger WSNs. However, those properties come with the trade-off that the maximum supported network size is not as large as the other schemes, where it sacrifices key connectivity to decrease the storage usage [2, 10].

References

- [1] Du W., Deng J., Han Y., Chen S., and Varshney P., "A Key Management Scheme for Wireless Sensor Networks using Deployment Knowledge". In IEEE Infocom'04, 2004.
- [2] Chan H., Perrig A., and Song D., "Random Key Pre-distribution Schemes for Sensor Networks". In IEEE Symposium on research in Security and Privacy, <http://www.cs.berkeley.edu/~dawnsong/papers/key-dist.pdf>, 2003.
- [3] Eschenauer L., and Gligor V., "A Key Management Scheme for Distributed Sensor Networks". In 9th ACM conference on Computer and Communications Security, <http://delivery.acm.org/10.1145/590000/586117/p41-eschenauer.pdf?key1=586117&key2=9921578121&coll=GUIDE&dl=GUIDE&CFID=81941059&CFTOKEN=46888212>, 2002.
- [4] Hwang D., Lai B., and Verbaughede I., "Energy-Memory-Security Tradeoffs in Distributed Sensor Networks". In 3rd International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW 2004), 2004.
- [5] Hwang J., and Kim Y., "Revisiting Random Key Pre-distribution for Sensor Networks". In ACM workshop on Security of Ad-Hoc and Sensor Networks (SASN 04), 2004.
- [6] "Key Distribution in Wireless Sensor Networks", http://en.wikipedia.org/wiki/Key_distribution_in_wireless_sensor_networks.
- [7] Liu D., and Ning P., "Location Based Pair-wise Key Establishment for Static Sensor Networks". In 1st ACM workshop on Security of ad-Hoc and Sensor Networks, 2003.
- [8] Merkle R., "Secure Communication over Insecure Channels". In Communications of the ACM, 1978.
- [9] Pietro R., Mancini L., and Mei A., "Random Key Assignment Secure Wireless Sensor Networks". In 1st ACM workshop on Security of ad-Hoc and Sensor Networks, 2003.
- [10] Seyit A., Yener B., "Key Distribution Mechanisms for Wireless Sensor Networks: A Survey". A technical report TR-05-07, Rensselaer Polytechnic Institute, <http://www.cs.rpi.edu/research/pdf/05-07.pdf>, 2005.
- [11] Zhu S., Xu S., Setia S., and Jajodia S., "Establishing pair-wise keys for secure communication in Ad-Hoc networks: A Probabilistic Approach". In 11th IEEE International Conference on Network Protocols (ICNP'03), 2003.