SGKP: Self Configured Group based Key Predistribution for Reducing Storage Overhead in Key Establishment for Wireless Sensor Networks

Manoj.R
Assistant Professor
Department of IT
Toc H Institute of science and Technology,Ernakulam

ABSTRACT
Sensor networks consist of a large number of sensors that monitor the environment, and a few base stations that collect the sensor readings. Security services such as authentication and key management [1] are critical to secure the communication between sensor nodes in hostile environments. Sensor nodes may establish keys between each other through key predistribution, where keying materials are predistributed to sensor nodes before deployment. Pre-distribution of secret keys for all pairs of nodes is not viable due to the large amount of memory used when the network size is large. If we know which nodes are more likely to stay in the same neighborhood before deployment, keys can be decided a priori. There exist a number of key pre-distribution schemes. A naive solution is to let all the nodes carry a master secret key. Any pair of nodes can use this global master secret key to achieve key agreement and obtain a new pairwise key[2,3]. Another key pre-distribution scheme[4] is to let each sensor carry N − 1 secret pairwise keys, each of which is known only to this sensor and one of the other N−1 sensors (assuming N is the total number of sensors).Symmetric key agreement is significant to security provisioning in sensor networks with resource limitations. A number of pairwise key predistribution protocols have been proposed, but the performance is often constrained by the unavailability of topology information before deployment and the limited storage budget within sensors. SGKP, a self-configured scheme for bootstrapping keys in large-scale sensor networks. SGKP is topology-adaptive, which requires no preloaded keying information but lets sensors compute shared keys with their neighbors after deployment.

General Terms
Security, Wireless Sensor Networks.

Keywords
Key predistribution, Self configured scheme, Pairwise Key, Service node, Worker node.

1. INTRODUCTION
In group based key predistribution [1] storage overhead is more, since sensors have to be preloaded with information. Self-configured scheme is used to reduce storage overhead. Self-configured scheme require no preloading of information but allow sensors to compute shared key with neighbors after deployment. Group-based schemes divide sensors into disjoint groups. Each sensor contains information for intra-group key establishment and inter-group key establishment. SGKP have better performance in terms of scalability, connectivity, resilience, and storage overhead.
Symmetric key agreement is significant to security provisioning in sensor networks with resource limitations. A number of pairwise key predistribution protocols have been proposed, but the performance is often constrained by the unavailability of topology information before deployment and the limited storage budget within sensors. SGKP, self configured group based key predistribution is a scheme for bootstrapping keys in large-scale sensor networks. SGKP is topology-adaptive, which requires no preloaded keying information but lets sensors compute shared keys with their neighbors after deployment.

Sensors differentiate their roles as either service nodes or worker nodes after deployment by probing the local connectivity of the network. Service sensors construct key spaces, and distribute keying information in order for worker sensors to bootstrap pairwise keys[2,3]. By exploring the λ-collusion resistant property of the key spaces, SGKP achieves perfect security in against node capture attack. Further, SGKP has low storage overhead but can establish secure communication for almost all pairs of neighboring sensors. SGKP is a localized procedure, which preloads no random information to any sensor before deployment, and thus achieves a perfect scalability in network size.

2. OVERVIEW OF SELF CONFIGURED GROUP BASED KEY DISTRIBUTION
The self configured group based key distribution can be represented as shown in Fig.1. The sensor node stores boot starp program which contains system parameters and key space and direct key establishment and path key establishment is done. Direct key established for nodes in the same group and path key is established for cross groups. Path key needs appropriate bridges to selected. Fig.2 describes how the secure key is obtained by sensor node. Service node selects key K.Worker node encrypts the key and sends in back to service node. Service node decrypts the information and sends back to the worker node.

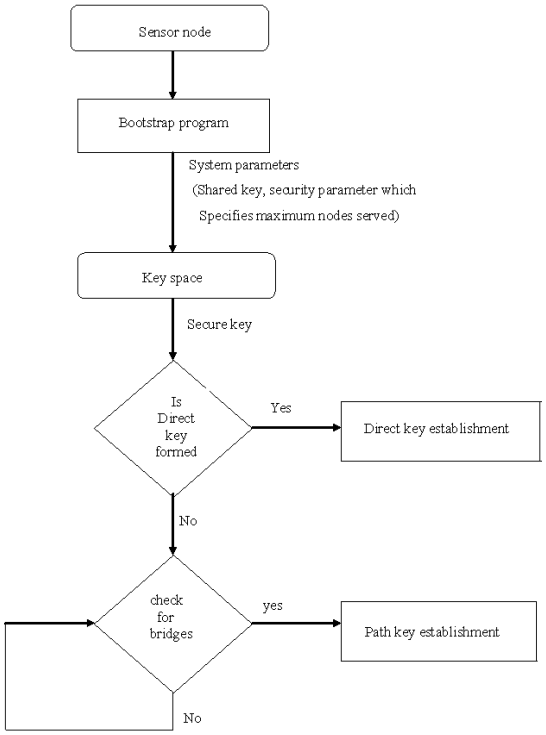

Fig 1: SKGP Overview

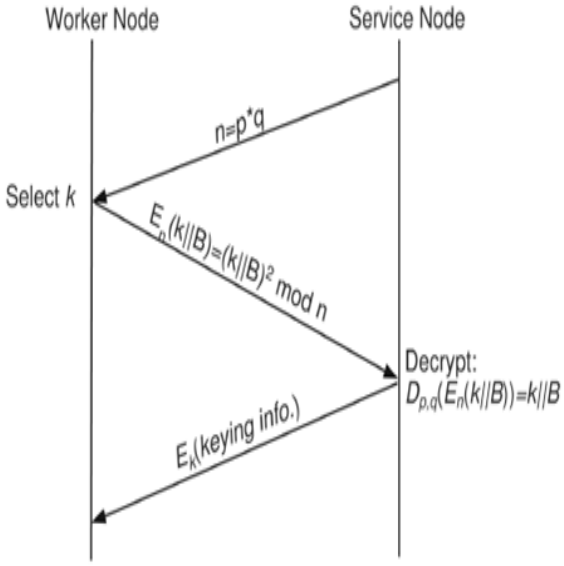

Fig 2: Sensor node obtaining secure key

2.1 Group Construction in SGKP

Sensor nodes to be deployed are divided into 'n' deployment groups $\{G_i\}i=1\ldots n$. For each deployment group G_i, a deployment instance D_i for pairwise key establishment. To handle pairwise key establishment for same group we construct in-group instance. In different deployment group, construct 'm' cross groups $\{G'_i\}i=1\ldots m$. Group G_i contain sensor nodes with ID as in equation (1).

$$ID = \{(i-1)m + j\}\ where\ j = 1,2,\ldots,n \qquad (1)$$

Cross group G'_i also contains certain sensor node. The ID's of the cross group sensor nodes (ID1)can be calculated using equation (2).

$$ID1 = \{i + (j-1)m\}\ where\ j = 1,2,\ldots,m \qquad (2)$$

where n - number of deployment groups, m - number sensor nodes in the same deployment group.

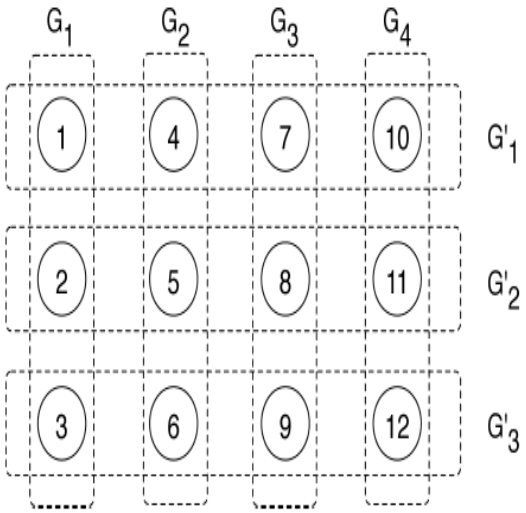

Fig 3: Group Construction in SKGP

Suppose For example if 'n' = 4 (deployment groups) and m = 3 (number of sensor nodes in a deployment group). To establish ID's, i = 1,2,3,4 and j = 1,2,3, deployment of 12 nodes for values of i and j are done using the equation (1). Group construction in SGKP is shown in Fig. 3.

2.2 Framework for Self configured group based key pre-distribution (SGKP)

A key pre-distribution technique can usually be divided into three different phases, the pre-distribution, which specifies how to predistribute keying materials to each sensor node, the direct key establishment, which specifies how to establish a pairwise key[3] shared between two sensor nodes directly, and the path key establishment, which specifies how to discover a sequence of sensor nodes to help two given sensor nodes to establish a temporary session key. A key established in the direct key establishment phase is called a direct key, while a key established in the path key establishment phase is called an indirect key.

2.2.1 Predistribution

Each sensor is preloaded with bootstrapping program that controls the role differentiation and node configuration. For sensor node that decides to become service node, this program computes the key space. Preconfigured parameters such as shared key by all sensors to secure the bootstrap procedure, security parameter that specifies the maximum number of nodes served by sensor. Sensor bootstraps elects itself as service node. SBK is topology-adaptive. Sensors differentiate their roles as either service nodes or worker nodes after deployment by probing the local connectivity of the network. Service sensors construct key spaces, and distribute keying information in order for worker sensors to bootstrap pairwise keys.

Before deployment, each sensor is preloaded with a bootstrapping program that controls the role differentiation and node configuration. For a sensor that decides to become a service node, this program also computes a key space and two

prime numbers for securely disseminate crypto shares to worker nodes in the vicinity.

Several pre-configured system parameters, are also uploaded to each sensor. *k0*, *akey* shared by all sensors, is employed to secure all the messages exchanged during the bootstrapping procedure. λ, the security parameter, determines the maximum number of worker nodes to be served by a service sensor.

2.2.1 Direct Key Establishment

After the predistribution step, each sensor node has the keying materials for two key predistribution instances, an in-group instance and a cross-group instance. Hence, the direct key establishment between two neighbor sensor nodes is simple and direct. If they are in the same deployment group, for example, G_i , they can follow the direct key establishment of the in-group instance D_i. If they are not in the same deployment group but belong to the same cross group G'_j they can follow the direct key establishment of the cross-group instance D'_j. To determine if two sensor nodes 'u' and 'v' are in the same deployment group or the same cross group, they only need to know the ID (either u or v) of the other party due to our group construction method.

2.2.1 Path Key Establishment

If two sensor nodes cannot establish a direct key, they have to go through the path key establishment to find a sequence of other sensor nodes to help them establish an indirect key. It is required that any pair of adjacent sensor nodes in this sequence can establish a direct key between them to make sure the security of key establishment. It is also assumed that every message between two sensor nodes is encrypted and authenticated by the direct key established between them.

When two sensor nodes belong to two different groups G_i and G_j, we use a different method to establish an indirect key. Basically, we need to find a 'bridge' between these two deployment groups in order to establish a cross group indirect key. A bridge between group G_i and G_j is defined as a pair of sensor nodes <a,b><a Є G_i and b Є G_j> that belong to the same cross group G'_k. A bridge is valid when the two sensor nodes involved in this bridge can establish a direct key. According to the predistribution step, there are 'm' potential bridges (one from each cross group) between every two deployment groups that can be used to establish an indirect key between sensor nodes in these two deployment groups. In addition, due to our group construction method, a sensor node can easily compute all possible bridges between any two deployment groups. Three bridges between group G1 and G4 are < 1, 10>, <2, 11>, <3, 12>

Path key establishment works as follows:

1. Source node u first tries to bridge involving itself to establish indirect key with destination node v. Bridge is <u,v'>.Node u send request to establish direct key with v' .node v' forwa request to v to establish indirect key.
2. If this step fails, node u try to bridge involving destination node. Bridge is <u',v>.node u send request to establish direct key with u' .node u' forward request to v to establish indirect key.
3. If both the above step fails, randomly use choose a bridge <u', v'>. Node u send request to establish direct key with u' .node u' forward request to v' to establish indirect key. If v' can establish direct key with v ,it forward the request to v to establish indirect key.

3. IMPLEMENTATION OF SGKP

The tool used to simulate is network simulator 2 (NS2)[5]. NS-2 is a Discrete Event Simulator that models network protocols like Wired, Wireless and satellite. It also simulates various transport layer protocols like TCP, UDP, multicast and unicast and various applications like Web, telnet, ftp.

Sensors are preloaded with a bootstrap program and several system parameters in the predistribution phase, and differentiate their roles as either service nodes or worker nodes in the node self-configuration phase. A worker sensor obtains keying information through a computationally asymmetric secure channel from a service node in the crypto share distribution phase, and then computes keys shared with those nodes belonging to the same key space in the shared key discovery phase.

Sensor nodes obtain secure key to establish a computationally asymmetric secure channel through which crypto shares can be delivered from a service sensor to a worker sensor. System requires a public key n and a private key (p, q) such that in equation (3).

$$n = p * q \qquad (3)$$

where p and q are large primes. The encryption of a message M, denoted by $E_n(M||B) = (M||B)^2 \bmod n$, where B is a predefined pattern for ambiguity resolution in decryption. However, recovering the plaintext M by computing $D_{p,q}(E_n(M||B))$ takes much higher computation. The direct key establishment is simulated as shown in figure 4. The path key establishment is shown in figure 5 and the service nodes providing keys to worker nodes is shown in figure 6.

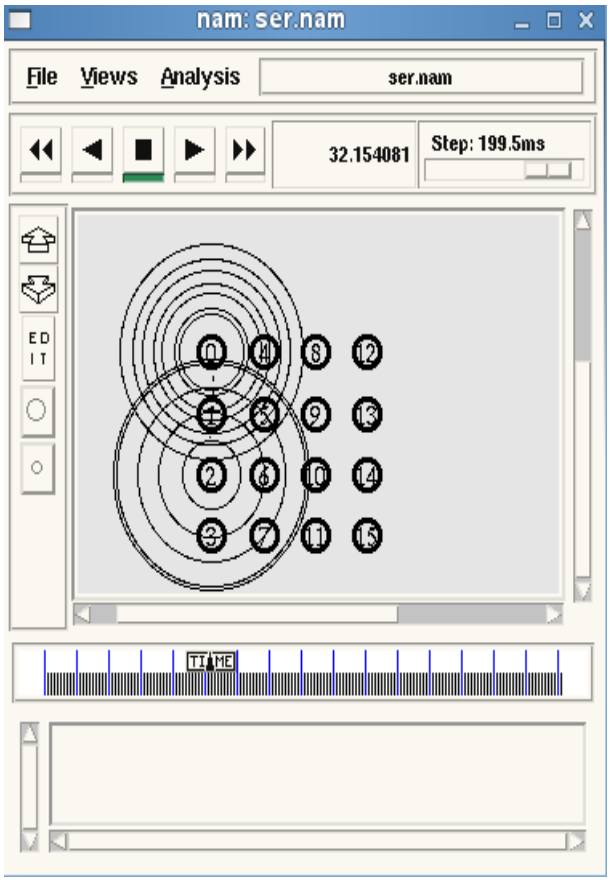

Figure 4(a) nodes establishing direct key

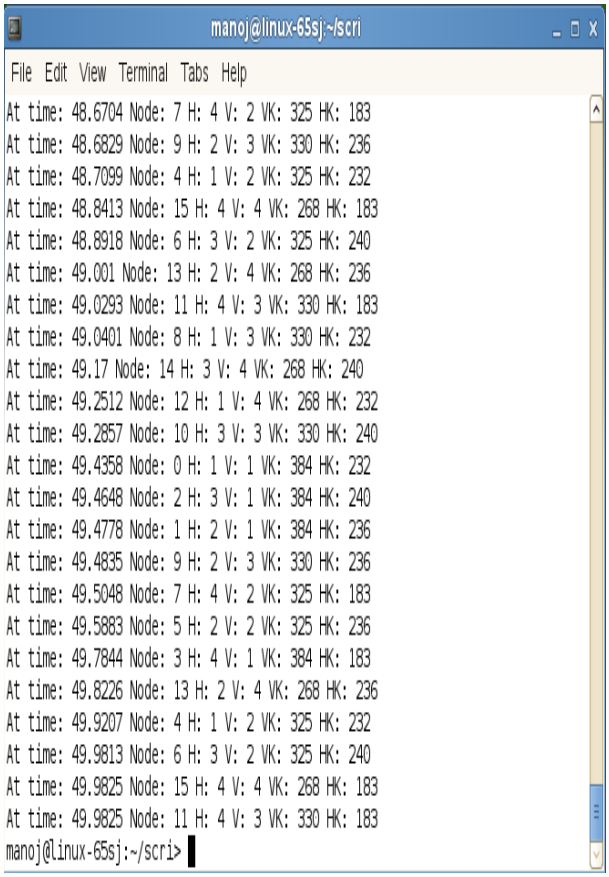

Figure 4.(b) Direct key Establishment

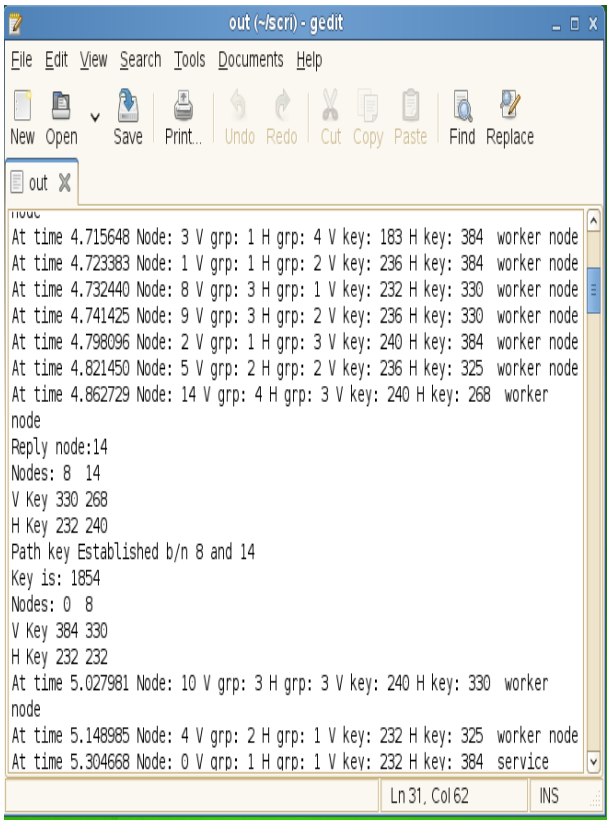

Figure 5. Path key establishment

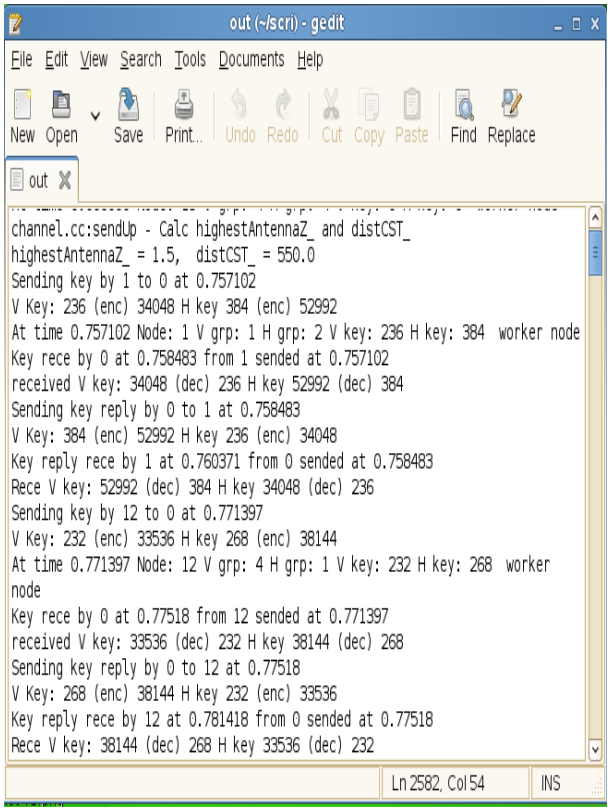

Figure 6. Service node provides keys to worker nodes

In group based key predistribution storage overhead is more, since sensors have to be preloaded with information. Self configured scheme is used to reduce storage overhead. Self configured scheme require no preloading of information but allow sensors to compute shared key with neighbors after deployment. The graph (Figure 7) shows that storage overhead is reduced by Self configured scheme.

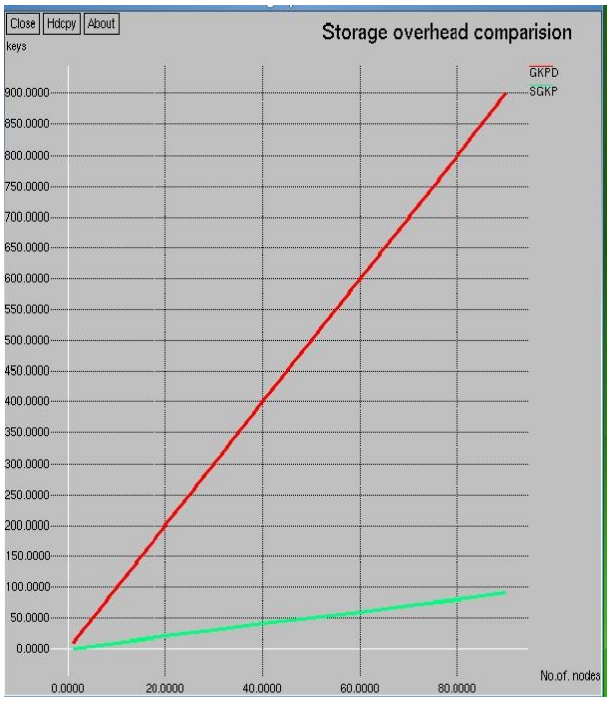

Figure 7. Storage overhead comparisons

4. CONCLUSION AND FUTURE SCOPE

Group-based schemes divide sensors into disjoint groups. Each sensor contains information for intra-group key establishment and inter-group key establishment. The simulation of Self configured Group based Key Predistribution is done by NS2. SGKP have better performance in terms of scalability, connectivity, resilience, and storage overhead.

As service node has to store the keys for all worker nodes, there is an increase in storage overhead and computational overhead on a single node. Service node has to store keys of all other node. If the number of nodes increases, then service node requires large amount of storage space. This makes an impact on service node's storage overhead. The solution for the problem has to be explored. The future scope is to find a method to distribute the service node and create access policies for keys.

5. REFERENCES

[1] Donggang Liu, Peng Ning and Wenlian Du, "Group-Based Key Predistribution for Wireless Sensor Networks," *ACM Transactions on Sensor Networks*,vol. 4, no. 2, Article 11,pp. 11-29, March 2008.

[2] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks,"*Proc. of the 10th ACM Conf. on Computer and Communications Security (CCS '03)*, pp. 52-61, 2003.

[3] P. Ning, R. Li, and D. Liu, "Establishing pairwise keys in distributed Sensor networks," *ACM Trans. on Information and System Security*, vol.8, no. 1, pp 41-77, February 2005.

[4] W. Du, J. Deng, Y. S. Han, S. Chen and P. K. Varshney, "A Key management scheme for wireless sensor networks using deployment knowledge ,"*Proc. of IEEE INFOCOM*, 2004.

[5] MarcGreis (2005). *Tutorial for the network simulator ns.* Retrieved from: http://www.isi.edu/nsnam/ns/tutorial/index.html