# Scalable Grouping Random Key Predistribution in Large Scale Wireless Sensor Networks

Po-Jen Chuang*, Tun-Hao Chao and Bo-Yi Li

*Department of Electrical Engineering, Tamkang University,*
*Tamsui, Taiwan 251, R.O.C.*

## Abstract

The security issue in a wireless sensor network (WSN) has been drawing considerable research attention in recent years. Key management, a basic security service, becomes the core design for various security services, such as encryption and authentication. To increase the connectivity of each key in a large-scale WSN and to enlarge its maximum supportable network size, this paper presents a scalable grouping (SG) random key predistribution scheme. The SG scheme divides all nodes into several groups and uses the one-way function to generate group-to-group pairwise keys. To improve resilience against node capture, i.e., to fortify the security strength, the scheme takes on the concept that the link key is composed of some shared keys. For any two nodes with two or more shared keys, the SG scheme uses the exclusive-OR operation to compose the link key -- assuring the link key used to secure a link is nearly unique. Experimental results show that the SG scheme is able to generate better resilience against node capture and higher scalability than existing random key based schemes.

***Key Words***: Wireless Sensor Networks (WSNs), Key Management, Random Key Predistribution, Performance Evaluation

## 1. Introduction

The development of wireless sensor networks has become an important research topic in recent years due to such critical application needs as emergency response, medical monitoring, military tracking, energy management and pollution monitoring.

A wireless sensor network (WSN) holds the following basic features:

- Low bandwidth and computing power
- Limited memory and energy resources
- Being prone to failure
- Large scale of sensor nodes
- Possibly without a central device

A sensor network can be easily assaulted or compromised by adversaries because it is often deployed in unattended environments. To enhance its security, researchers have come up with a number of security services, including key management [e.g., 1−5] – our investigation focus in this paper. A key management protocol for WSNs should be simple and light due to limited processing power, battery life, communication bandwidth and memory space of the sensor nodes. In this sense, the random key predistribution scheme [6] which allows each node to communicate only with its neighbor nodes is appropriate for large scale WSNs. (Note that traditional key exchange mechanisms in wired networks, such as the Diffie-Hellman key agreement [7] and the public key cryptography [8], are not fit for WSNs because they usually involve high computational overhead and considerable memory requirement. Key distribution protocols relying on infrastructures or trusted third parties are also impractical in WSNs considering their restricted communication ranges and inability to learn about the network topology before deployment.)

There are two extreme cases of key predistribution.

(1) Each node stores a single master key.

*Corresponding author. E-mail: pjchuang@ee.tku.edu.tw

(2) Each node stores all the other nodes' pairwise keys.

In case (1), each node consumes only limited memory space, but the entire network will be disrupted when an adversary compromises a node. Case (2) may provide the highest security (as the key every two nodes use to communicate with each other is unique) but is nevertheless unfeasible considering the limited memory resource of a node. For instance, if $n$, the number of nodes in the network, is large, it will be difficult for each node to store $n - 1$ keys.

Based on the concept of the random key predistribution, this paper presents a scalable grouping (SG) random key predistribution scheme to increase the connectivity of each key in large-scale WSNs and the maximum supportable network size. The proposed SG scheme, a modification of the unique assigned one-way (UAO) function scheme [9], divides all nodes into several groups and uses the one-way function to generate group-to-group pairwise keys. To assure that the link key used to secure a link is nearly unique, the SG scheme uses the exclusive-OR operation to compose the link key when two nodes have two or more shared keys. Experimental evaluation is conducted to compare the performance of the proposed SG scheme and other random key based schemes, and the results exhibit better performance for our SG scheme in terms of resilience against node capture (security strength) and supportable network sizes (scalability).

## 2. Previous Random Key Based Schemes

Background investigation into several previous random key based schemes is provided in this section to facilitate later discussion. This background investigation focuses only on the random key based key predistribution schemes which assume the location information of each node unknown and unpredictable before deployment. Schemes (such as [10]) which divide the deployment region into several sub-regions and assign keys to nodes according to the sub-regions they are to be deployed to or schemes (such as [11,12]) which further predict the topology of the network and assign the pairwise keys to the predicted neighboring node pairs are not of our focus in this research and therefore left out from further discussions.

### 2.1 The Random Key Predistribution Scheme [8]

In the random key predistribution scheme (referred to as the basic scheme in this paper), each node randomly chooses a fixed number of keys before being deployed and after deployment establishes secure links with neighbors having shared keys. For two neighbor nodes having no shared keys, a path-key will be established in-between. The basic scheme covers the following three phases.

(1) The key pre-distribution phase: Before deployment, each node picks r keys from a large key pool of S and stores them into its memory to form a key ring.

(2) The shared-key discovery phase: Each node tries to find out if its neighbors share a key with it. If a shared key exists between the node and a neighbor, it becomes the link key for transmission between the two nodes and a graph is gradually formed.

(3) The path-key establishment phase: For two neighbor nodes having no shared keys, establish a path-key through other secure links (as Figure 1 indicates).

The random key predistribution scheme uses the random graph theory [13] to analyze suitable parameters. Consider a random graph with $n$ nodes $G(n, p_l)$ and assume the probability that a link exists between any two nodes is $p_l$. When $p_l = 0$, the graph has no edges; when $p_l = 1$, the graph is fully connected. We find in [13] the monotone property that there exists a threshold value of $p_l$ and that the property will move from "likely false" to "likely true" on whether graph $G$ is connected. The threshold function $p_l$ is defined as

$$p_l = \frac{\ln(n) - \ln(-\ln(P_c))}{n}$$

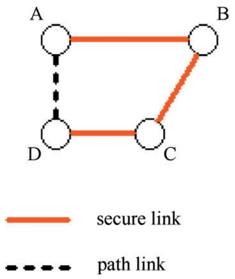

**Figure 1.** Establishing a path-key.

where $P_c$ is the desired probability that graph $G$ is connected. The expected number of secure links for a node will be

$$d = p_l \times (n-1) = \frac{(n-1)\times(\ln(n) - \ln(-\ln(P_c)))}{n}$$

With the expected degree $d$ and the average number of neighbor nodes for each node, say $n'$, we can obtain the required probability $p$ for any two neighbor nodes to establish a secure link successfully

$$p = \frac{d}{n'}$$

Note that probability $p$ is used to evaluate the performance of our proposed SG scheme and other random key based schemes in this paper. In general, the value of $n'$ will be greater than that of $d$. If $n'$ is less than $d$, $p$ will be larger than 1 – which makes achieving the desired $P_c$ unlikely even if every node is able to set up secure links with all its neighbor nodes.

### 2.2 The q-Composite Keys Scheme [14]

The q-composite keys scheme works similarly as the basic scheme. It adopts a parameter $q$ to enhance security against smaller-scale attacks. At least $q$ shared keys are required to establish a secure link between any two nodes and the link key $K$ is generated as the hash of all shared keys:

$K$ = hash($K1$ || $K2$ ||…|| $Kq'$), where $q' \geqq q$.

As the probability for any two nodes to share a minimum of $q$ keys may be less than the required probability $p$, the key pool size needs to shrink until the probability of connectivity equals $p$.

The q-composite key scheme can strengthen network security only when the number of compromised nodes is limited. When the number of compromised nodes grows beyond the threshold, the scheme becomes less secure. When q = 1 (i.e., when the number of keys needed to construct a secure link = 1), the q-composite keys scheme becomes the the random key predistribution scheme. In this special case, the two schemes hold the same key pool size. However, when a node has two or more keys, the 1-composite keys scheme will compose them into a new key and thus yield lower compromised rates than the basic scheme.

### 2.3 The Random-Pairwise Keys Scheme [14]

The basic idea of the random-pairwise keys scheme is to store in each node sufficient pairwise keys, instead of all pairwise keys, of the other nodes to form a connected graph. Assume each node stores at most $m$ keys and the required probability for two neighbor nodes to set up a secure link is $p$. The maximum supportable network size $n$ can be calculated as

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

The operation of the scheme includes two phases.

(1) The initialization phase: A total of $n = m/p$ unique node identities are generated. As the actual size of the network may be smaller than $n$, unused node identities will be used when additional nodes are added to the network. Each node identity is matched up with $m$ other randomly selected distinct node IDs. A pairwise key is generated for each pair of nodes and is stored in the key rings of both nodes, along with the ID of the other node that also knows the key.

(2) The key-setup phase: Each node first broadcasts its ID to the immediate neighbors. A node will find out if it shares a common pairwise communication key with a neighbor node by searching for each other's IDs in the key ring. A cryptographic handshake is performed between neighbor nodes who wish to mutually verify that they do indeed have knowledge of the key.

The random-pairwise keys scheme is able to provide node-to-node authentication properties as it uses pairwise keys instead of picking up keys from a large key pool. The fact that each key used to secure links is unique helps the scheme produce more desirable resilience against node capture than the basic scheme and the q-composite keys scheme. The better performance is nevertheless obtained at the cost of larger memory space and a restriction on the number of newly added nodes.

## 2.4 The Unique Assigned One-Way (UAO) Function Scheme [9]

The UAO scheme also uses pairwise keys derived from a unique one-way function in each node to establish the secure link. It can support larger networks than the random-pairwise keys scheme due to smaller required node memory: Only one side of the link needs to store the key; the other side can obtain the key through the one-way function.

The UAO scheme will perform a key decision algorithm before deploying nodes. Assume that each sensor node $SN_i$ has a unique identifier $ID_i$ and is assigned a unique one-way function $F_i$. Each node first randomly selects $r$ node identifiers ($r$ is the required number of keys) to achieve the connected graph, calculates $r$ pairwise keys $K_j$ by equation $K_j = F_j(ID_i)$ ($j$ being the selected node identifier), and then memorizes $r$ pairs of $K_j$ and $ID_j$. After node deployment, each node performs the node-to-node authentication protocol to set up secure links with its neighbors. An $SN_i$ first broadcasts its $ID_i$ to all neighbor nodes who then verify if the received $ID_i$ is combined with any key in their key rings. If a neighbor node $SN_s$ finds a key in its key ring, say $K_s$, combined with the received $ID_i$, it will send a request message encrypted by $K_s$ and its own identifier $ID_s$ to $SN_i$. After receiving the message, $SN_i$ obtains the key $K_s$ by computing $F_i$ ($ID_s$). Both nodes ($i$ and $s$) verify the link key $K_s$ through the challenge-response process and establish a secure link between them.

To obtain the maximum supportable network size of the scheme, let $p$ be the probability for any two neighbor nodes to set up a secure link, $n$ be the size of the network, and every node store $r$ keys from $n - 1$ pairwise keys. The probability for a node to have a specific key will be $r/(n - 1)$, and the probability that the node does not have the specific key will be $1 - [r/(n - 1)]$. As two neighbor nodes can construct a secure link when one of them has the pairwise key, we get the probability that two neighbor nodes have no shared keys as $\{1 - [r/(n - 1)]\}^2$, and the probability for the two nodes to construct a secure link will be

$$p = 1 - (1 - \frac{r}{n-1})^2$$

The UAO scheme is shown in [9] to support larger network sizes than the random-pairwise keys scheme. As to security, both schemes are as efficient in preventing nodes from intruding attacks.

# 3. The Scalable Grouping Random Key Predistribution Scheme

The scalable grouping (SG) random key predistribution scheme is proposed in this paper to support more nodes and to provide desirable resilience against node capture in a sensor network. As mentioned in Section 2, the random-pairwise keys scheme achieves the highest security at the cost of large node memory space, and the UAO scheme tries to reduce such memory requirement by using the one-way function to assist the forming of link keys and attains as favorable security as the random-pairwise keys scheme (because each key used to secure a link is unique).

Different from the UAO scheme, our proposed SG scheme divides all nodes into several groups to increase the connecting ability of each key and the maximum supportable network size. It also takes on the concept that the link key is composed of some shared keys to improve resilience against node capture. Listed below are the unique features of the $k$-SG scheme, $k$ being the maximum number of nodes in a group.

- Each node in the network has a group identifier.
- Each group has at most $k$ nodes.
- Nodes of the same group have a shared group key.
- Two nodes of different groups use the group-to-group pairwise key to establish the secure link.
- If two nodes have two or more shared keys, the link key is the composite of these shared keys using the exclusive-OR operation.

The operation of the $k$-SG scheme consists of three phases.

(1) **Initialization**: The initialization phase runs offline. We first divide all nodes into several groups, each group having at most $k$ nodes. All nodes of the same group $i$ have a group identifier $GID_i$, a group key $K_i$ and a one-way function $F_i$. Each node in the group then randomly selects $r$ group identifiers ($r$ being the required number of keys to achieve the connected graph), calculates $r$ group pairwise keys $K_{ji}$, ($j$ being the selected group identifier $GID_j$) and

stores $r$ pairs of $K_{ji}$ and $GID_j$. The equation for generating group-to-group pairwise keys is.

**(2) Link key setup**: After deployment, sensor nodes work to set up link keys. Each sensor node (in group $i$) starts by broadcasting its $GID_i$ to the neighbor nodes and meanwhile receiving group identifiers from neighbors. If a node in group $i$ finds out through GID verification that it is in the same group with a neighbor, the node then returns the message (of being in the same group) and the list of its key ring to that neighbor. Otherwise (i.e., the neighbor node is in a different group), the node will return its group identifier $GID_i$ to the neighbor. In either case, both nodes move on to execute the link key setup algorithm. Assuming the group identifiers of nodes A and B are respectively $GID_i$ and $GID_j$, the link key setup algorithm can be defined as follows.

```
if ($GID_i$ = $GID_j$){ /* node A belongs to the same group as node B */
    if (A has other shared keys with B)
        Link Key $K_s = K_i \oplus$ shared keys;
    else
        $K_s = K_i$;
}
else { /* node A and node B are not in the same group */
    switch (){
        case 1 (A has $K_{ji}$ and B has $K_{ij}$):
            A calculates $K_{ij} = F_i(GID_j)$;
            B calculates $K_{ji} = F_j(GID_i)$;
            $K_s = K_{ij} \oplus K_{ji}$;
            break;
        case 2 (A has $K_{ji}$ but B does not have $K_{ij}$):
            B calculates $K_{ji}$, then $K_s = K_{ji}$;
            break;
        case 3 (B has $K_{ij}$ but A does not have $K_{ji}$):
            A calculates $K_{ij}$, then $K_s = K_{ij}$;
            break;
        case 4 (A does not have $K_{ji}$ and B does not have $K_{ij}$):
            A can not set up a secure link with B;
            break;
    }
}
```

To give an example, Figure 2 gives three neighbor nodes A, B and C (respectively with group identifiers $GID_1$, $GID_1$ and $GID_3$) and their key rings. As we can see, the link key ($K_s$) generated by nodes A and B is composed of a group key $K_1$ and a shard key $K_{21}$, and the link key generated by nodes A and C is the group-to-group pairwise key $K_{31}$ set up by node C using $F_3(GID_1)$.

**(3) Secure link establishment**: After execution of the link key setup algorithm, any two nodes with a shared link key $K_s$ will verify $K_s$ through the challenge-response protocol. If the result turns out correct, a secure link is then established between the two nodes for communication.

## 4. Performance Evaluation

Simulation runs have been conducted to evaluate the performance of the proposed $k$-SG scheme ($k = 2$ and 3), the basic scheme and the q-composite keys scheme (q = 2 and 3). The performance parameters of interest include security strength, maximum supportable network sizes under limited memory resources and conformity to the limited global payoff requirement. The following are some necessary notations.

***n***: the number of sensor nodes in the network

***n'***: the average number of neighbor nodes for each node

***r***: the key ring size (the number of keys in each node)

***p***: the probability for two neighbors to set up a secure link

***k***: the group size of the $k$-SG scheme (the maximum number of nodes for each group)

***g***: the number of groups in the network (= $n/k$)

***X***: the number of directly compromised links (i.e., links

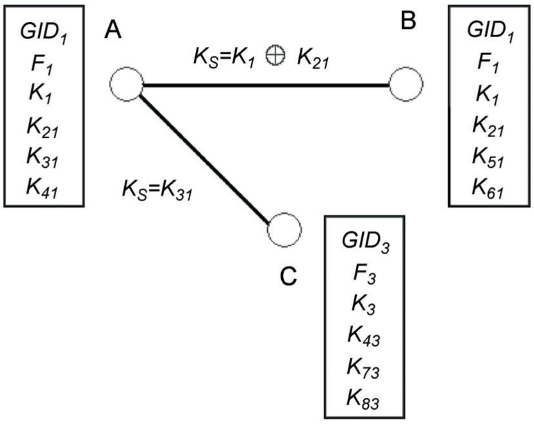

**Figure 2.** A link key setup example of the SG scheme.

with compromised node(s))

*Y*: the number of indirectly (or additionally) compromised links (i.e., links with no compromised nodes but their link keys can be recovered by adversaries)

### 4.1 The Security Strength

The security of the SG scheme is evaluated in terms of **resilience against node capture**. The fraction of links in a network which an adversary is able to eavesdrop indirectly (by recovering keys from the captured nodes) is estimated as

$$\frac{Y}{number\ of\ links - X}$$

and can be calculated following these steps.

- Randomly pick up *i* nodes (*i* is the expected number of compromised nodes).
- Place all group keys and group-to-group pairwise keys stored in the *i* nodes into the adversary's database.
- Put all one-way functions stored in the *i* nodes into the adversary's database -- all keys that can be derived from these functions will be considered as compromised keys.
- For each link which is not directly compromised, check its link key to see if it can be recovered from the database.

The key ring size (*r*) for each scheme is listed in Table 1. (The key ring for the *k*-SG scheme includes an extra key -- the group key.)

Figures 3 and 4 give the resilience against node capture under different numbers of sensor nodes compromised for the schemes. Note that both the random-pairwise keys scheme and the UAO scheme are presented by y = 0 as adversaries can not get global information from local node capture.

**Table 1.** The key ring size for each scheme ($p = 0.33$)

| | $n = 1000$ | $n = 2000$ |
|---|---|---|
| the basic scheme | 200 | 200 |
| 2-composite keys scheme | 200 | 200 |
| 3-composite keys scheme | 200 | 200 |
| the 2-SG scheme | 90 + 1 | 180 + 1 |
| the 3-SG scheme | 60 + 1 | 120 + 1 |

Figure 3 shows that among the schemes the 2-SG scheme yields the largest security strength when $n = 1000$. This is because the proposed SG scheme adopts the group key and the group-to-group pairwise key, instead of picking up keys from a large key pool. Thus whenever an adversary compromises a node, it can get the information about other nodes in the same group only, significantly reducing the fraction of additional compromised links.

When the network size extends to $n = 2000$ in Figure 4, the SG scheme provides even stronger security because in large-sized networks the ratio of information obtained by an adversary to that of the whole network decreases.

### 4.2 The Maximum Supportable Network Size

For schemes using pairwise keys to establish secure

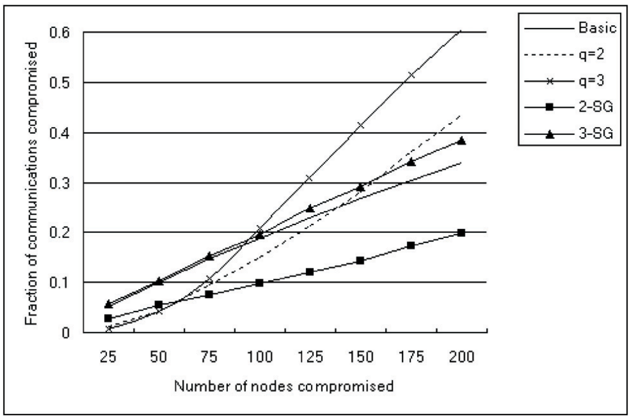

**Figure 3.** The number of compromised nodes versus the fraction of additional compromised links for various schemes ($n = 1000$, $n' = 60$, $p = 0.33$).

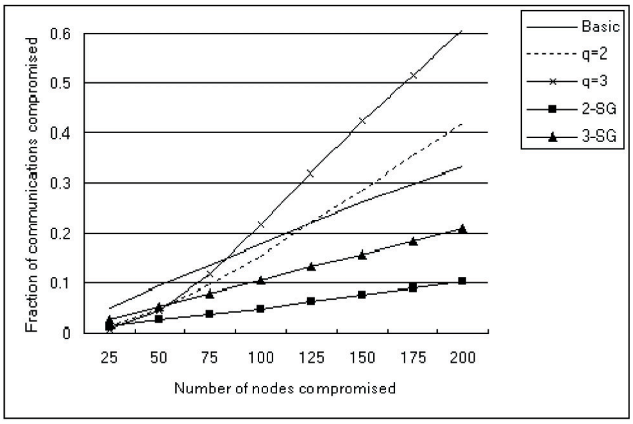

**Figure 4.** The number of compromised nodes versus the fraction of additional compromised links for various schemes ($n = 2000$, $n' = 60$, $p = 0.33$).

links, the key ring size directly determines the probability for two neighbor nodes to set up a secure link. Scalability is hence restricted. Take the random-pairwise keys scheme as an example. Assuming that the network size $n$ = 10000 and the probability of connectivity $p = 0.33$, the required key ring size $r$ will be $r = n \times p = 3300$, which is quite impractical in a sensor network.

Now suppose $p_a/p_b$ is respectively the probability for two neighbor nodes with the same group identifier/different group identifiers to set up a secure link. In the SG scheme, $p$ can be represented by the following equation

$$p = \frac{1}{g} \times p_a + \left(1 - \frac{1}{g}\right) \times p_b$$

where $1/g$ is the probability for any two nodes to have the same group identifier, $p_a = 1$ (because the two nodes have a shared group key) and $p_b$ = 1 – the probability that neither node has the key derived from the other's one-way function. As the key ring size = $r$, the probability for any node to get a key derived from a particular node's one-way function will be $r/(g - 1)$. Thus

$$P_b = 1 - \left(1 - \frac{r}{g-1}\right)^2$$

The probability of connectivity $p$ in the SG scheme can now be derived as

$$p = \frac{1}{g} + \left(1 - \frac{1}{g}\right) \times \left[1 - \left(\frac{r}{g-1}\right)^2\right] \qquad (1)$$

or be simplified by $g = n/k$ into

$$p = \frac{k}{n} + \left(1 - \frac{k}{n}\right) \times \left[1 - \left(1 - \frac{rk}{n-k}\right)^2\right] \qquad (2)$$

Equation (2) is adopted to evaluate the maximum supportable network size for the $k$-SG scheme. The maximum supportable network sizes for various schemes are plotted in Figure 5 under $p = 0.33$, key length = 128 bits and the size of the one-way function = 160 bits.

The result displays that the SG scheme supports the largest-sized network (due to its unique grouping feature) and that the maximum supportable network size grows when $k$ increases. It is however more practical to keep $k < 4$ due to the limited global payoff requirement (to be discussed later).

Now consider the SG scheme at the time when nodes are newly added into the network. Assume that r = 200, the maximum supportable network size of the 2-SG scheme is 2210 nodes and the network has at most 1105 groups. In this case, 1105 GIDs are generated before deployment and then assigned to all ready-to-deploy nodes. Those unassigned at this phase will get assigned when new nodes are added to the network at a later time.

### 4.3 The Limited Global Payoff Requirement

The limited global payoff requirement [14] is applied to the SG scheme because its secure links can be indirectly compromised, like the basic scheme and the q-composite scheme. The main purpose of the requirement is to keep the adversary from gaining too much at little expense. Take the basic scheme as an example. In a network with size $n$ = 10000, when an adversary compromises 50 nodes, there will be about 9.5% additional compromised links. As the ratio of compromised nodes to the whole network size is only 0.5%, random key based schemes need to meet the requirement in [14] to avoid such a situation.

- The number of additional compromised links in the network ≦ the number of directly compromised links in the network (i.e. Y ≦ X).

The limited global payoff ratio ρ is thus defined to be

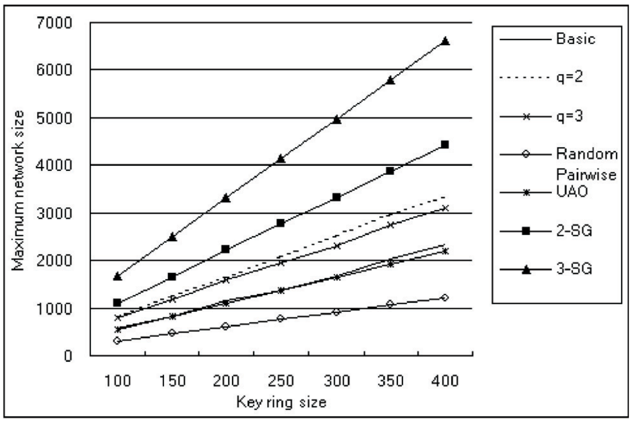

**Figure 5.** The maximum supportable network sizes for various schemes ($p = 0.33$, the key length = 128 bits and the size of the one-way function = 160 bits).

Y/X and the limited global requirement will be satisfied when ρ ≦ 1.

The limited global payoff ratios for the 2-SG scheme and 3-SG scheme are respectively given in Figures 6 and 7. Figures 6(a), (b) and (c) evaluate X and Y under different network sizes while (d) calculates the limited global payoff ratio ρ. As the result indicates, the 2-SG scheme constantly meets the requirement (ρ ≦ 1) regardless of the key ring size and the number of compromised nodes. By contrast, Figure 7 shows the 3-SG scheme turns over ρ around 1 and up, barely meeting the requirement (although it supports the largest network size among all schemes). Note that we do not discuss the k-SG scheme with k > 3 because in such cases the value of ρ always exceeds one – which apparently disagrees with the limited global payoff requirement.

## 5. Conclusion

The security issue in a wireless sensor network (WSN) has been drawing considerable research attention in recent years. Key management, a basic security service, becomes the core design for several security services, including encryption and authentication. To increase the connectivity of each key in a large-scale WSN and to enlarge its maximum supportable network size, this paper presents a scalable grouping (SG) random key predistribution scheme. The proposed SG scheme divides all nodes in a WSN into several groups and uses the one-way function to generate group-to-group pairwise keys. To improve resilience against node capture, i.e., to fortify the security strength, the scheme takes on the concept that the link key is composed of some shared keys. For any two nodes with two or more shared keys, the SG scheme uses the exclusive-OR operation to compose the link key -- assuring the link key used to secure a link is nearly unique. Experimental results exhibit that our proposed scheme maintains the largest security strength (better resilience against node

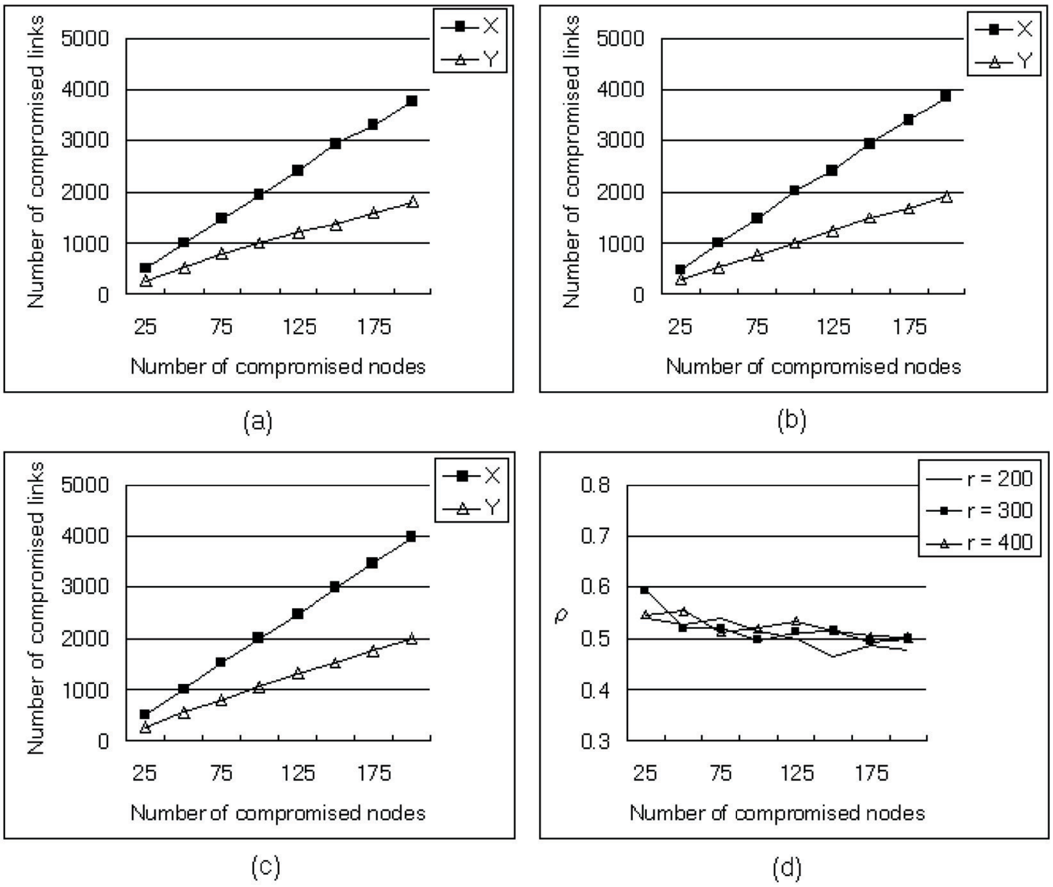

**Figure 6.** The limited global payoff requirement for the 2-SG scheme under (a) $n = 2210$, (b) $n = 3312$ and (c) $n = 4414$. The ratio ρ is calculated in (d).

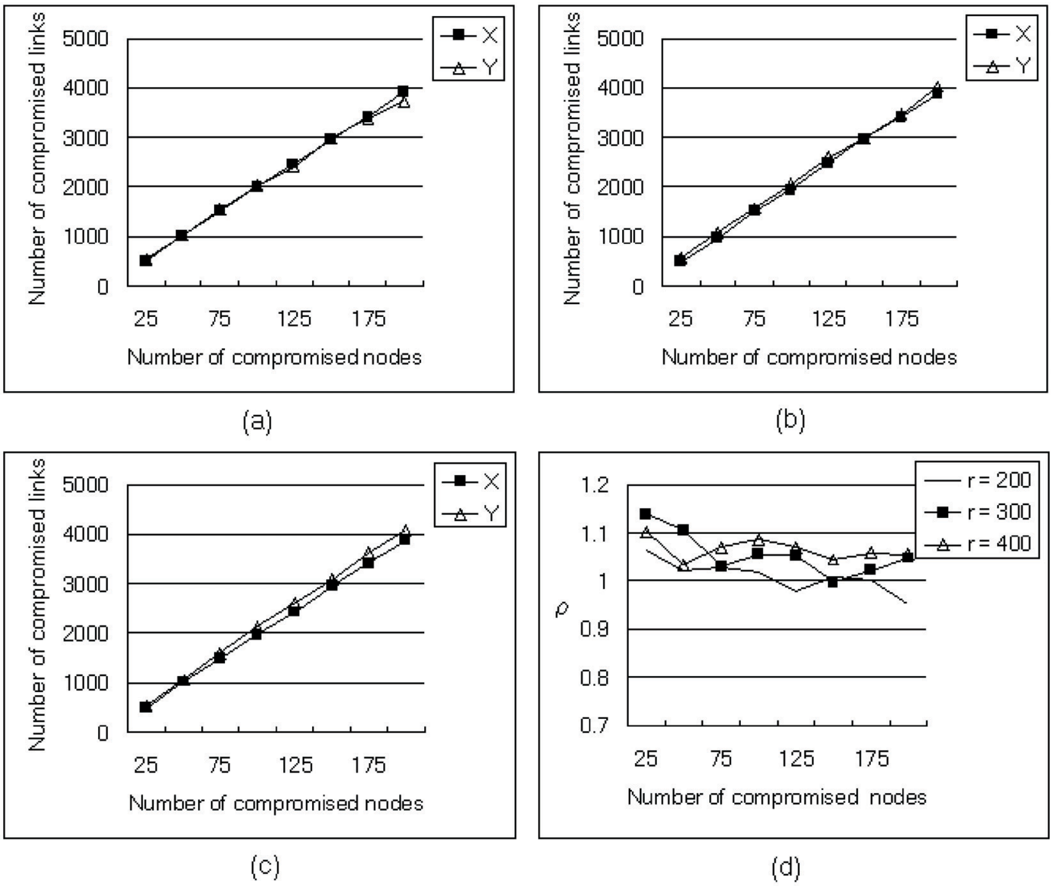

**Figure 7.** The limited global payoff requirement for the 3-SG scheme under (a) $n = 3315$, (b) $n = 4968$ and (c) $n = 6621$. The ratio ρ is calculated in (d).

capture) than other random key based schemes because it adopts the group key and group-to-group pairwise keys instead of picking up keys from a large key pool. Due to its unique grouping feature, the SG scheme is also shown to yield higher scalability, i.e., to support larger maximum supportable network sizes.

## Acknowledgment

This work was supported in part by the National Science Council, Taiwan, R. O. C., under Grants No. NSC 93-2213-E-032-032 and No. NSC 97-2221-E-032-018. A preliminary version of this paper was presented at the 3rd International Conference on Information Technology and Applications, July 2005.

## References

[1] Perrig, A., Szewczyk, R., Wen, V., Culler, D. and Tygar, J. D., “SPINS: Security Protocols for Sensor Networks,” *Proc. 7th Annual ACM Int’l Conf. on Mobile Computing and Networks,* July, pp. 189–199 (2001).

[2] Huang, Q., Cukier, J., Kobayashi, H., Liu, B. and Zhang, J., “Fast Authenticated Key Establishment Protocols for Self-Organizing Sensor Networks,” *Proc. 2nd ACM Int’l Conf. on Wireless sensor networks and applications,* Sep., pp. 141–150 (2003).

[3] Zhu, S., Setia, S. and Jajodia, S., “LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks,” *Proc. 10th ACM Conf. on Computer and Communication Security,* Oct., pp. 62–72 (2003).

[4] Liu, D. and Ning, P., “Establishing Pairwise Keys in Distributed Sensor Networks,” *Proc. 10th ACM Conf. on Computer and Communications Security,* Oct., pp. 52–61 (2003).

[5] Wadaa, A., Olariu, S., Wilson, L. and Eltoweissy, M., “Scalable Cryptographic Key Management in Wireless Sensor Networks,” *Proc. 24th Int’l Conf. on Dis-*

*tributed Computing Systems Workshops,* Mar., pp. 796–802 (2004).

[6] Diffie, W. and Hellman, M. E., "New Directions in Cryptography," *IEEE Trans. Inform. Theory,* Vol. 22, pp. 644–654 (1976).

[7] Rivest, R. L., Shamir, A. and Adleman, L. M., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM,* Vol. 21, pp. 120–126 (1978).

[8] Eschenauer, L. and Gligor, V. D., "A Key-Management Scheme for Distributed Sensor Networks," *Proc. 9th ACM Conf. on Computer and Communication Security,* Nov., pp. 41–47 (2002).

[9] Wu, S. Y. and Shieh, S. P., "Adaptive Random Key Distribution Schemes for Wireless Sensor Networks," *Proc. Int'l Workshop on Advanced Developments in Software and Systems Security,* Dec. (2003).

[10] Jiang, Y. and Shi, H., "A Cluster-Based Random Key Pre-Distribution Scheme in Large Scale Sensor Networks," *Proc. 3rd IEEE Int'l Conf. on Natural Computation,* Vol. 2, pp. 462–466 (2007).

[11] Canh, N. T., Lee, Y. K. and Lee, S., "HGKM: A Group-Based Key Management Scheme for Sensor Networks Using Deployment Knowledge," *Proc. 6th Annual IEEE Conf. on Communication Networks and Services Research Conference,* May, pp. 544–551 (2008).

[12] Du, W., Deng, J., Han, Y. S. and Varshney, P. K., "A Key Predistribution Scheme for Sensor Networks Using Deployment Knowledge," *IEEE Trans. on Dependable and Secure Computing,* Vol. 3, pp. 62–77 (2006).

[13] Spencer, J., "The Strange Logic of Random Graphs," *Algorithms and Combinatorics 22,* Springer-Verlag (2000).

[14] Chan, H., Perrig, A. and Song, D., "Random Key Predistribution Schemes for Sensor Networks," *Proc. 2003 IEEE Symp. on Security and Privacy,* May, pp. 197–213 (2003).

***Manuscript Received: May. 21, 2007***
***Accepted: Sep. 24, 2008***