# A Grid-based Key Pre-distribution Scheme Using Multiple Keyspace Pools For Wireless Sensor Networks

Yu-Kun Ho
*Department of Electrical Engineering National Cheng Kung University, Tainan, Taiwan, R.O.C*
ykho@eembox.ee.ncku.edu.tw

Chia-Yu Liu
stan517@gmail.com

## Abstract

*Because of resource constraints and large scale environment, it is not easy to achieve that establishing the communicating key in wireless sensor network. And traditional key exchange and key Distribution protocols are not practical. There are many key agreement schemes used in general network, such as RSA and other public-key based schemes, but they are not suitable for WSNs due to the limited computational abilities of the sensor nodes.*

*We present an Grid-based scheme for key pre-distribution, and the key pool is divided into multigroups as multiple keyspace pools. Each sensor nodes acquire keyspace form the corresponding row and column keyspace pool according to their coordinates. And it will calculate a key from their same keyspace by combining the symmetric key generation scheme. The concepts and analysis of this scheme compare with existing scheme in resilience to node capture. There can be very high probability and efficiency to establish the communicating key between sensor nodes. We also present an in depth analysis of our scheme in terms of network resilience and associated overhead.*

## 1. Introduction

The wireless sensor networks (WSN) have been developed rapidly by the advances in hardware and wireless network technologies in recently years [7][17]. These tiny sensor nodes, which consist of sensing, data processing, and communicating components, leverage the idea of sensor networks. Sensor networks represent a better improvement over traditional sensors. Sensor networks often contain one or more sinks that provide centralized control. A sink typically serves as the access point for the user or as a gateway to another network. Wireless sensor networks are scattered by hundreds of sensor nodes for dealing with some popular tasks such as environment monitoring, scientific measurement, battlefield surveillance, security and disaster management.

A lot of works have been made in recent years to develop the application for wireless sensor networks, such as coverage [10], routing protocol [14], target tracking [5], data fusion [3], etc.

It is as stated above, the wireless sensor network is in opening area, its privacy and reliability is worse than the traditional network. And it is very easy to be attacked from outside adversary. In sensor network security, an important challenge is the design of protocols to establish a secure communications from a collection of sensor nodes. Therefore it is a quite important researching subject to guarantee the integrity and the safety of data transmitting. Due to the limited computational and power constraints of sensor nodes, key distribution schemes such as public key algorithm [21] including Diffie-Hellman key agreement or RSA [2] signature are impractical.

A naive solution is single mission-key. It is to let all nodes store an identical master secret key. Any pair of nodes can use this master secret key to establish a new pairwise key securely. However, if a single node is compromised, the security of the entire sensor networks is compromised. Some existing studies suggest storing the master key in tamper-resistant hardware to reduce the risk. At the other extreme, one might consider a key pre-distribution scheme in which each sensor stores N-1 keys (where N is the number of nodes), it requires N(N-1) keys per node. This scheme guarantees perfect resilience because compromised nodes do not leak information about keys shared between two non-compromised nodes. Unfortunately, this scheme is too expensive for sensors with an extremely limited amount of memory because N may be very large. There are other schemes, such as combinational design [16], hash function [6], polynomial share [4].

Recently, Eschenauer and Gligor proposed a key management scheme [9] based on probability key sharing and utilization of a simple shared-key

discovery protocol for key distribution, key revocation, and node re-keying.

### 1.1 Key distribution

Key distribution consists of three parts, namely key pre-distribution, share-key discovery, and path-key establishment. The proposed scheme focuses on this key distribution phase.

Before sensor networks deploying, the key pre-distribution part initially generates a large key pool of keys. Each sensor node receives a key ring with a randomly chosen subset of keys from the key pool. This part makes that only a small number of keys need to be placed on each sensor node's key ring. Therefore any two nodes share at least a key with a chosen probability. Figure 1 shows an example that each sensor node receives m keys randomly from a Key Pool.

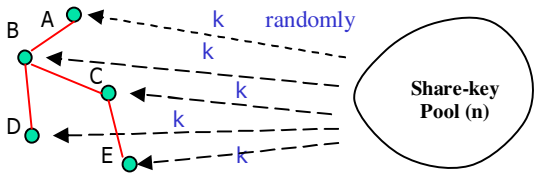


**Figure 1: nodes acquire key randomly from a shared-key pool**

The shared-key discovery part takes place during wireless sensor network initialization in the operational environment. And every node discovers its neighbors in wireless communication range with which it shares keys. In short, any two nodes take a share-key discovery if they share a key. Therefore it will establish the topology of the sensor field by the routing layer of the WSN. A link exists between two sensor nodes only if they share a common key, and if a link exists between two nodes, all communications on that link is secured by encryption. Because the key rings of keys chosen randomly from the same pool, it is possible that the same key is shared by more than a pair of sensor nodes. And the adversary has no opportunity to attack in this part.

The path-key establishment part assigns a path-key to selected pairs of sensor nodes in wireless communication range that do not share a key but are connected by two or more links.

### 1.2 Revocation

When a sensor node is compromised, it is necessary to be able to revoke the whole key ring of the compromised nodes. And a controller node will broadcast a revocation message containing a signed list of k key identifiers for the key ring to be revoked. Once the keys are removed from key rings, some links may disappear, and the affected nodes need to reconfigure those links by restarting the shared-key discovery and, possibly path-key establishment.

### 1.3 Re-keying

It is possible that the WSN may refresh once or twice, the lifetime of keys expires and re-keying must take place. Re-keying is equivalent with a self-revocation of a key by a node. As such, it does not involve any network-wide broadcast message from a controller, and it is simpler. After removing the expired-key, the affected nodes restart the shared-key discovery and the path-key establishment possibly.

The remainder of this paper is organized as follows. Section 2 describes the backgrounds and related research. Section 3 describe our scheme and analyze the resilience against node capture, and Section 4 compares our scheme with existing key pre-distribution schemes. Finally we conclude all in Section 5.

## 2. Background

We will introduce the related frameworks and mechanisms of our scheme in this section. They include grid-based of sensor networks [8], [19], [20], [23], and symmetric key generation system [15], [22].

### 2.1 Grid-based sensor networks

Because the amount of sensor nodes may be a large number in WSNs, it is difficult and diseconomy to deploy regular or special topologies in advance. We can adopt the grid-based framework to allocate every sensor position. Figure 2 shows an environment of typical wireless sensor networks in grid-based. Each grid size is $\alpha \times \alpha$. S is a sink, B is a source, and others are normal sensor nodes. Although it may cause a little error of sensor position, the influence is much less in key distribution phase of WSNs.

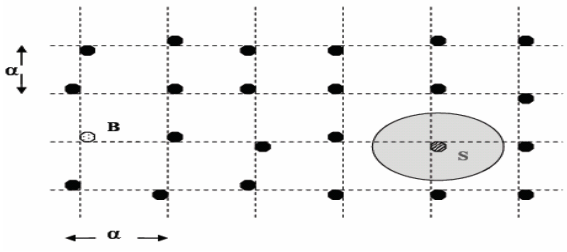


**Figure 2: Typical grid-based sensor networks**

Sadi and Kim [19] proposed a polynomial key pre-distribution method based on grid. According to figure 3, there are 2mω polynomials in a finite field. For each sensor, the setup server selects an unoccupied intersection (i, j) in the grid and assigns it to the node. So, the ID of this sensor is <i, j>.Then the setup server distributes ID, and 2τ polynomials to the sensor node. Here 2τ polynomials are randomly chosen from $i^{th}$ row and the remaining τ from $j^{th}$ column.

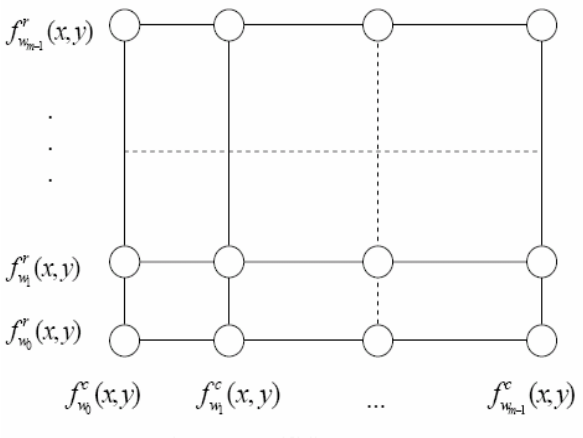


**Figure 3: a grid-base wireless sensor network with 2mω polynomials**

### 2.2 Symmetric key generation system

Recently, Blom[15] proposed a symmetric key generation system that allows any pair of nodes in networks to be able to get a pairwise key. And it has a property that is λ-secure. As long as no more than λ nodes are compromised, the network is perfectly secure. We simply introduce Blom's method and describe how it works in key pre-distribution phase.

Initially, it constructs a (λ + 1) × N matrix G over a finite field GF(q) first, where N is the size of the network. G is considered as public information. Matrix G can be considered as follows: (s is regarded as seed.) [22]

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ s & s^2 & s^3 & \cdots & s^N \\ s^2 & (s^2)^2 & (s^3)^2 & \cdots & (s^N)^2 \\ & & \vdots & & \\ s^\lambda & (s^2)^\lambda & (s^3)^\lambda & \cdots & (s^N)^\lambda \end{bmatrix}$$

After that the base system generates a random (λ+1) × (λ+1) symmetric matrix D over GF(q), and computes an N × (λ + 1) matrix $A = (D \bullet G)^T$. Matrix D is private, not public. Because D is a symmetric matrix, we can know the relations by following inferring:

$$A \bullet G = (D \bullet G)^T \bullet G = G^T \bullet D^T \bullet G = G^T \bullet D \bullet G = (A \bullet G)^T .$$

We assign $K = A \bullet G$, so we can also understand K is a symmetric matrix. It means that $K_{ij} = K_{ji}$ in matrix K. If node i and node j want to communicate to each other, it will use $K_{ij}$ (or $K_{ji}$) as the pairwise key. And it uses the following distributing algorithm:

For K = 1 to N:
  $Node_K$ stores the $k^{th}$ row of matrix A;
  $Node_K$ stores the $k^{th}$ column of matrix G;
End;

Figure 4 shows how the pairwise key ( $K_{ij} = K_{ji}$ ) is generated. Thus, when nodes i and j need to communicate to each other, they have to find the pairwise key between them, they first exchange their columns of G, and they use their private rows of A to compute $K_{ij}$ and $K_{ji}$ respectively. [22]

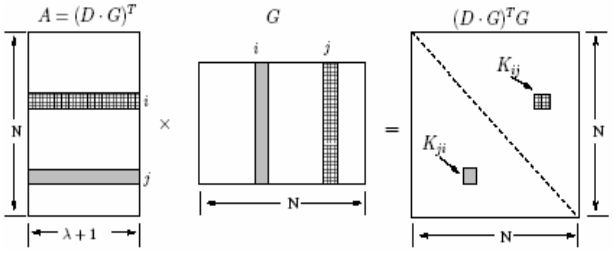


**Figure 4: generation of computing pairwise key**

## 3. Key Pre-distribution Scheme by Using Multiple Keyspace Pools

Since it may be asymmetrical and unequal to distribute all sensor nodes randomly, we construct the wireless sensor networks on grid-based. And to reduce the computation of key generation, we replace the polynomial shares with keyspace pools in our scheme. The keyspace pools consist of several computed matrices that we meant last section. Before describing our proposed scheme, we first define that a sensor node select one keyspace (*D*, *G'*) if the node carries the secret information generated from (*D*, *G'*) using Blom's symmetric key generation scheme.

### 3.1 Key Pre-distribution phase

According to Figure 5, there are N sensor nodes in WSNs. We construct m × m grid to cover it, where $m = \sqrt{N}$ .We first initialize 2m keyspace pools corresponded to each row and column in the grid, and we assign numbers to them. respectively: {$KS_{r1}$, $KS_{r2}$,..., $KS_{rm}$ , $KS_{c1}$, $KS_{c2}$,..., $KS_{cm}$}.

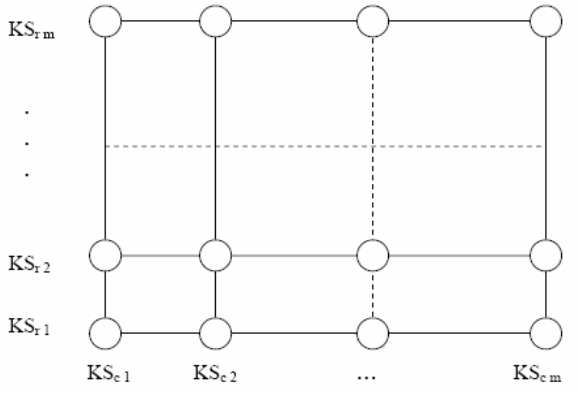


**Figure 5: m × m grid and 2m keyspace pools**

Before computing the private key, we should distribute some information in every node. Our pre-distribution phase contains three steps:

#### Generating matrix G:

First, we choose a element to create a generator matrix $G$ of size $(\lambda+1) \times N$ from a finite field $GF(q)$, where $q > N$. Let $G(j)$ represent the $j^{th}$ column of $G$, and provide $G(j)$ to node $j$. Although $G(j)$ consists of $(\lambda+1)$ elements, each sensor just needs to store one seed to reconstruct all elements in matrix G. Since the seed is unique for every node, we also regard it as the ID of a node.

#### Generating matrix D:

In 2m keyspace pools, each pool generates $\omega$ symmetric matrices $\{D_1,. . ., D_\omega\}$of size $(\lambda + 1) \times (\lambda + 1)$. Therefore there are $2m\omega$ symmetric matrices in full.We then compute the matrix $A_i = (D_i \cdot G)^T$, where $A_i(j)$ represent the $j^{th}$ row of $A_i$.

#### Allocating matrix to each node:

For each node, we randomly select $2\tau$ distinct matrices from the two keyspace pools corresponded to the vertical and horizontal according to the coordinates of the nodes. Here is a simple algorithm for allocating matrices.

For ID = 0,1…N
  $Node_N$ whose coordinate is (i,j) who picks $2\tau$ matrices randomly from $KS_{ri}$ and $KS_{cj}$;
End;

Therefore, each node has to store $2\tau(\lambda+1)$ elements in its memory. And each node can use its own information to generate a communicating key.

### 3.2 Key generation phase

After distributing information to nodes, we first check their coordinates of nodes if two nodes want to communicate to each other. Assume $c_i$ is $i^{th}$ column and $r_j$ is $j^{th}$ row in the grid. To establish a pairwise key with node $j$, node $i$ checks whether $c_i = c_j$ or $r_i = r_j$. If $c_i = c_j$, both nodes share keyspace from $KS_{ci}$ and they can establish a key from using Blom's symmetric key scheme. Finally they can compute a pairwise key if they possess a common keyspace (matrix D). Similarly, if $r_i = r_j$ they can compute with the same way. By broadcasting messages, we can receive ID of nodes and regard as a seed of matrix G. Then we can use the seed to construct the matrix G. Since matrix D is random, we can compute the equation:

$$K_{ij} = K_{ji} = A_k(i) \cdot G(j) = A_k(j) \cdot G(i). \quad (1)$$

For example, $Node_i$ and $Node_j$ share a common keyspace that is $D_k$ after exchanging its own information. Then $Node_i$ and $Node_j$ use $D_k$ and seed s to compute their pairwise key from above equation.

### 3.3 Path-Key establishment phase

This phase is the time when $c_i \neq c_j$ and $r_i \neq r_j$ or when nodes have no common keyspace. If they are, they will have to search an extra node as an intermediate node.

.The intermediate node has common keyspace to establish communicating key with both the nodes $i$ and $j$. Since our environment is based on grid, it is easy to find that there is a node at least that can be taken as an intermediate node between any two sensors. We show an example as following figure 6. For node (i,i) and node (j,j) in the grid, there exists node(i,j) and node(j,i) that can be as intermediate node.

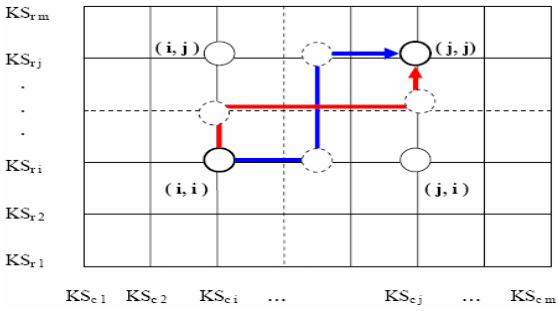


**Figure 6: an example of path-key establishment**

After checking the two nodes, we have to process multiple path-key discoveries if there is still no common keyspace in them. In multiple path-key discoveries, we try to look for other nodes that have a common keyspace by row and column sequentially. We can see it in blue color and red color path. To reduce the computation, we use linear search scheme to find the node. (For other special cases, the search mechanism can be modified.) Because of common row or column keyspace pool, it gets higher probability to share the same keyspace.

Figure 7 is a flow chart that describes how path-key phase works.

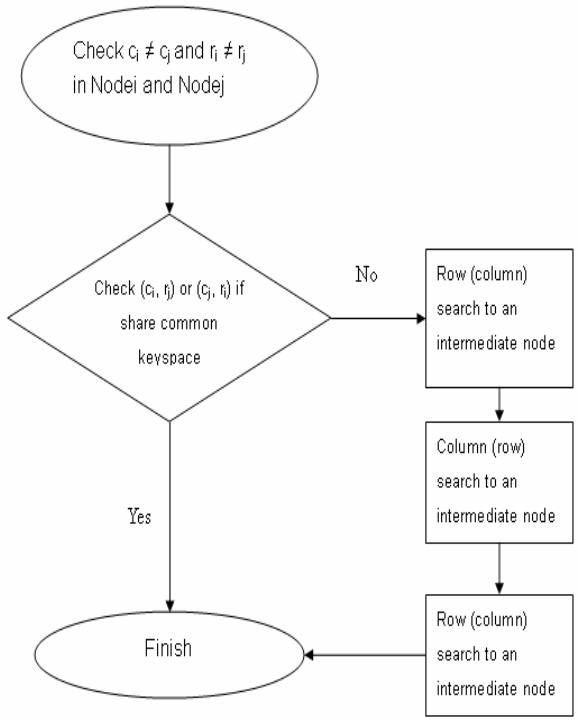


**Figure 7: a flow chart of path-key establishment**

### 3.4 Dealing with the added nodes

There may be many circumstances in WSNs to increase or decrease amount of nodes. Therefore, it has to re-distribute the information after that. In our scheme, we can use the following equation to finish it.

$$N_{after} = N_{before} + N_{new} \quad (2)$$

We can see that it is simple to deal with the added nodes, and we still can hold on the connectivity of the network.

### 3.5 Connectivity of the network

According above description, we can understand that each node chooses keyspaces from its row KS or column KS (KS is as keyspace pool in short.) There are 2(m-1) nodes that can establish pairwise key directly for each node in the grid. Therefore the probability of establishing pairwise key for each node is

$$P_{global} = \frac{2(m-1)}{N-1} \approx \frac{2(m-1)}{m^2-1} = \frac{2}{m+1} \quad (3)$$

For the nodes in a single row or column, each node picks $\tau$ matrices from $\omega$ keyspaces($\tau < \omega$). Therefore we know that the probability of establishing pairwise for each node in a single row or column is

$$P_{local} = 1 - \frac{\binom{\omega}{\tau}\binom{\omega-\tau}{\tau}}{\binom{\omega}{\tau}^2} \quad (4)$$

### 3.6 Memory usage

First, we define the unit of memory size as the size of a secret key. According to Blom's scheme, if a keyspace is $\lambda$-secure property, each node needs to take memory of size $\lambda + 1$ to store the keyspace information. Therefore, we consider the memory usage $M$ is

$$M = 2\tau(\lambda + 1) \quad (5)$$

Then from above equation, we see that the $\lambda$ *is*

$$\lambda = \left\lfloor \frac{M}{2\tau} \right\rfloor - 1 \quad (6)$$

### 3.7 Resilience

According to Blom's scheme, our scheme is also

with $\lambda$-secure property. Therefore our scheme is perfect secure if no more $\left\lfloor \frac{M}{2\tau} \right\rfloor - 1$ nodes are compromised.

We assume that an adversary randomly compromises *Nc* nodes. And the probability of a keyspace being chosen for a sensor node is $\frac{\tau}{m\omega}$.

And the probability *P(k)* of this keyspace being chosen exactly k times among Nc compromised sensor nodes is

$$P(k) = \binom{N_c}{k} \left(\frac{\tau}{m\omega}\right)^k \left(1 - \frac{\tau}{m\omega}\right)^{N_c - k} \quad (7)$$

Then the probability $P_{NC}$ of any keyspace being compromised is

$$P_{Nc} = 1 - \sum_{t=0}^{k} P(t) \quad (8)$$

Because of $\lambda$-secure property of our scheme, the resilience *P* of the network is

$$P = P_{Nc} - P_{\lambda} \quad (9)$$

According to above analysis, we see the resilience of our scheme related to the parameters which are ω、τ、m .We will simulate the mechanism that we proposed by using these parameters in next section.

## 4. Simulation Results

In this section, we simulate the analyses in last section. Our simulating tool is on GNU Octave version 2.1.73 (i686-pc-cygwin).

### 4.1 Probability of establishing a pairwise key

According to Equation (4) in section 3.5, we simulate this result with different ω and τ in figure 8.It is simple to understand that the two parameters ω and τ will affect the $P_{global}$ directly. The larger τ is or the lower ω is, the higher probability of establishing a pairwise key is.

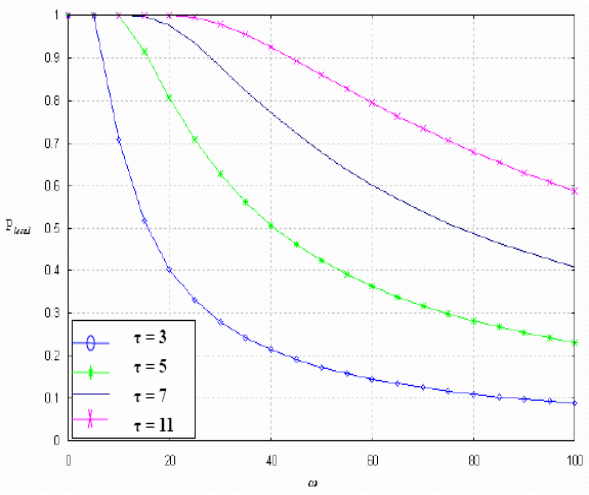


**Figure 8: Probability of establishing a pairwise key with different ω and τ.**

### 4.2 Result of path-key establishment

From section 3.3, we know that it may be no common keyspace between two nodes and they have to process path-key establishment. Following figure 9 is the result to this phase.

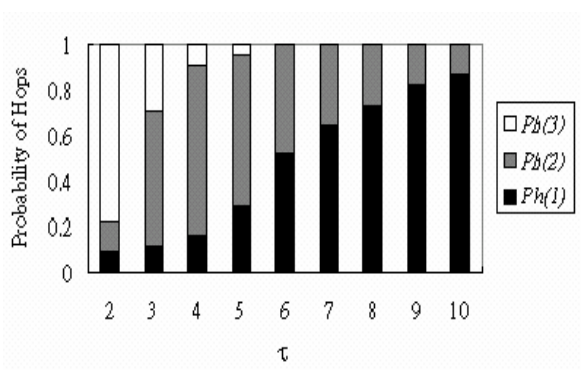


**Figure 9: Simulation of path-key establishment**

### 4.3 Resilience of our scheme

According to section 3.7, we see that the less nodes are compromised, the better resilience of mechanism is. We take the simulation with parameter that is

N = 4000: the numbers of sensor nodes.

M = 200: memory size of each sensor node.

We compare the resilience of ω = 5 with the resilience of ω = 11 in figure 10.

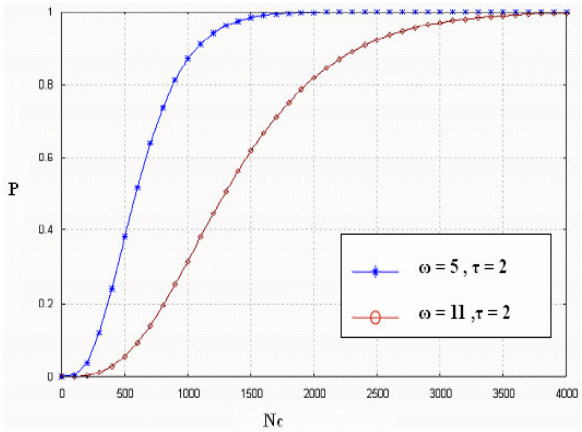


**Figure 10: comparing the resilience of ω = 5 with the resilience of ω = 11 (τ = 2).**

According to figure 10, we can see that $P$ = 0.9 of ω = 5 is worse than $P$ = 0.3 of ω = 11 in $Nc$ = 1000 nodes. Therefore, the larger ω is, the better resilience of our scheme is.

### 4.4 Comparing with other schemes

We regard our scheme as MKP scheme (Multiple Keysapce Pools scheme), and regard basis scheme as EG scheme (Eschenauer and Gligor scheme) [9].We show the comparison in figure 11.

The parameters in the simulation are that N = 9000, M= 200, ω = 11, and τ = 2.

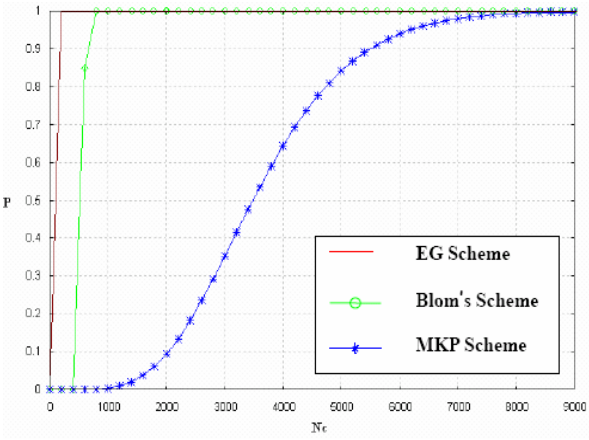


**Figure 11: Comparison of three schemes**

According to the figure 11, we can see the curves of our scheme raise more slowly. It means that the resilience of our scheme is better than others.

## 5. Conclusion and Future Work

Our scheme is based on grid for uniform distributing, and we also reduce computation by using simple mathematics. And our scheme gets better resilience finally.

It is another subject to extend our scheme, such as 3-D grid, path-key establishment, and some special cases.

## 6. References

[1] A. Perrig, J. Stanlovic, and D. Wagner, "Security in wireless sensor networks," Communications of the ACM, Vol. 47, pp. 6, June 2004.

[2] C. Blundo, A. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," Lecture Notes in Computer Science, pp. 1-23,1993.

[3] C.Intanagonwiwat, R Govindan, D Estrin, J.S. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," IEEE/ACM Transactions on Networking, vol. 11, pp. 2-16, 2003.

[4] D. Liu, P. Ning, and R. LI, "Establishing pairwise keys in distributed sensor networks," Trans. Information and System Security, Vol. 8, pp. 41-77, 2005.

[5] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," IEEE Signal Processing Magazine, pp. 61-72, 2002.

[6] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," Proc. IEEE Security and Privacy Symposim, pp.197-213 ,2003.

[7] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "A survey on sensor networks," IEEE Communications Magazine, vol. 40, pp. 102-114, 2002.

[8] K. Chakrabarty, S. Sitharama Iyengar, H. Qi, and E. Cho, "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks," IEEE Transaction On Computers, vol. 51, pp. 12, December 2002.

[9] L. Eschenauer, and V.D. Gligor, "A key-management scheme for distributed sensor networks," Proc. 9th ACM conference on Computer and Communications Security, pp. 41-47, 2002.

[10] M. Cardei, and J. Wu, "Coverage in Wireless Sensor Networks," Handbook of Sensor Networks. CRC Press, 2004.

[11] M. Hall, Combinatorial Theory,Wiley-Interscience in Discrete Mathematics,1986.

[12] M.J.B. Robshaw, "Security Estimates for 51 2-bit RSA," WESCON/'95. Conference record. 'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies', pp. 409, Nov. 1995.

[13] P. Erdös, and A. Rényi, "On random graphs," Publicationes Mathematicae, vol. 6, pp. 290-297, 1959.

[14] Q. Jiang, and D. Manivannan, "Routing Protocols for Sensor Networks," Proc. IEEE CCNC, pp. 93-98, 2004.

[15] R. Blom, "An optimal class of symmetric key generation systems," Proc. Eurocrypt, pp. 335-338 1984.

[16] S. Camtepe, and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor networks,"Proc. 9th European Symposium On Research in Computer Security, pp. 293-308, 2004.

[17] S. Tilak, N.B. Abu-ghazaleh, and W. Heinzelman, "A Taxonomy of Wireless Micro-Sensor Network Models," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 6, pp. 28-36, Apr. 2002.

[18] S. Zhu, S. Setia and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," Proc. 10th ACM Conference on Computer and Communications Security, pp. 62-72, Oct. 2003.

[19] Sadi, M.G.; Dong Seong Kim; Jong Sou Park,"GBR: grid based random key predistribution for wireless sensor network," Parallel and Distributed Systems, Proceedings. 11th International Conference, vol. 2, pp. 310-314, July 2005.

[20] V. Hingne, A. Joshi, E. Houstis, and J. Michopoulos, "On the Grid and Sensor Networks," Proceedings of the Fourth International Workshop on Grid Computing, pp. 166-173, Nov. 2003.

[21] W. Diffie, and M. E. Hellman, "New directions in cryptography," Trans. Information Theory, vol. 22, pp.644-654, Nov. 1976.

[22] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, and A. Khalili "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks, " Trans. Information and System Security, vol. 8, pp. 228-258, 2005.

[23] Z. WU, H. Song, S. Jiang, and X. Xu, "A Grid-based Stable Routing Algorithm in Mobile Ad Hoc Networks," Proceedings of the First Asia international Conference on Modelling & Simulation, pp. 181-186, March 2007.