# Weakness of Key Predistribution Scheme Proposed by J. Dong et al.

*Anupam Pattanayak, B. Majhi*

Computer Science & Engineering Department,
National Institute of Technology, Rourkela, India – 769008
{ **anupam.pk@gmail.com, bmajhi@nitrkl.ac.in** }

**Abstract.** A Sensor Node in Wireless Sensor Network has very limited resources such as processing capability, memory capacity, battery power, and communication capability. When the communication between any two sensor nodes are required to be secured, the symmetric key cryptography technique is used for its advantage over public key cryptography in terms of requirement of less resources. Keys are pre-distributed to each sensor node from a set of keys called key pool before deployment of sensors nodes. Combinatorial design helps in a great way to determine the way keys are drawn from the key pool for distributing to individual sensor nodes. J. Dong et al proposed a key predistribution scheme based on orthogonal array. We present the weakness of this predistribution scheme.

**Keywords:** Key predistribution, Combinatorial Design, Orthogonal Array, Resiliency

## 1. Introduction

In general, a sensor node consists of four basic units: (i) Processing unit, (ii) Sensing unit, (iii) Transceiver unit, and (iv) Power unit [6]. In general, a sensor network consists of a large number of sensor nodes. Sensor networks consisting of 10,000 nodes or 1,00,000 nodes are not uncommon. Although individual sensor nodes have limited resources, they are capable of achieving worthy task of big volume when they work as a group. Sensor networks are used in a number of different areas such as military, industry, health, environment, and home [3, 4]. Locations of sensor nodes in the network are not pre determined in most of the cases. This permits us to deploy sensor nodes in hostile environments such as border area of a hostile neighbor country by some means, for example, by using aeroplane.

When secured communication between two sensor nodes is required, then they can follow either symmetric key cryptography or asymmetric key cryptography. *Asymmetric key cryptography* requires huge computing resources that a tiny sensor can't afford. So a *symmetric key cryptography* is preferred. But *key generation and distribution* using *Diffie-Hellman key exchange protocol* or *public key infrastructure* is also more or less infeasible in distributed sensor network consisting of resource limited sensor nodes. So distributing a set of keys to each sensor node before their *deployment* is a good solution.

A sensor node can communicate with other node if the second one is lying within the circle of radio frequency range of the first one, and if both of them share a common key. In a key predistribution scheme, there are three phases. These are *key predistribution*, *shared-key discovery*, and *path-key establishment* [9]. In key predistribution phase a set of keys are chosen from a key pool following a predetermined order. In shared-key discovery phase if two nodes want to communicate and they are lying in one another's radio frequency range then they communicate to each other and find which one is the common key between them. If two nodes $N_i$ and $N_j$ want to communicate with each other and there is no common key then they look for one or more intermediate nodes such that every pair of adjacent nodes share a common key, so that $N_i$ and $N_j$ are able to communicate with each other securely. This phase is known as path-key establishment. In our survey we focus on the key predistribution phase only.

Key predistribution scheme (KPS) proposed by Camtepe and Yener was the first proposal that uses combinatorial design for key predistribution in distributed wireless sensor network [7, 8]. They have used two kinds of designs – projective plane and generalized quadrangle. Next Lee and Stinson proposed a KPS that uses transversal design [9]. Dong, Pei, and Wang have proposed two different kind of KPS based on Orthogonal Array [12], and based on 3-design [13]. The KPS proposed by Ruj and Roy is based on partially balanced incomplete block design (PBIBD) [10].

## 2. Background

The subject combinatorial design found its application initially in the design of experiments in statistics for long days. Then it was used extensively in the field of coding theory. Also combinatorial design techniques have been used in numerous other area of computer science such as Boolean function, authentication code, visual cryptography, multiple accesses to channel, software testing etc [1, 2].

### 2.1 Combinatorial Design

Combinatorial design theory is interested in arranging elements of a finite set into subsets to satisfy certain properties. It is the study of families of subsets with various prescribed regularity properties. Members of the universal set S in a combinatorial design are usually called *treatments*, or *varieties,* and the subsets chosen are called *blocks*. A *regular design* based on a v-set S is a collection of k-sets from S such that every member of S belongs to r of the k-set blocks. It is usual to write b for the number of blocks in a design. So a regular design has four parameters: $v$, $b$, $r$, and $k$. However these parameters are not independent. A regular design is represented as ($v$, $b$, $r$, $k$)-design.

In any regular design, $b*k = v*r$.

A block design is proper if all of its block have same length. The number of blocks that contain a given treatment is the replication number, $r$. If all v treatments occur in a block of a design, then the block is called *complete*. If a regular design has the property this property then that design is called *complete design.* Complete design is of very little

interest unless some further structure is imposed (such as in *Latin Square*). Given a design if at least one block is incomplete then the design is *incomplete* design. If v = b, the design is called *symmetric*.

If x and y are any two different treatments in an incomplete design, we shall refer to the number of blocks that contain both x and y as the *covalency* of x and y, and write it as $\lambda_{xy}$. Many important designs are concerned with this covalency function. More stuff on this interesting subject can be found in [14, 15].

### 2.2 **Orthogonal Array**

An N x K array A with elements from S (|S|=s) is said to be orthogonal array (OA) with s levels, strength t and index $\lambda$ ($0 \leq t \leq k$) if every N x t sub-array of A contains each t-tuple based on S exactly $\lambda$ times as a row.

When $\lambda$=1 we say OA has *index unity*.

N, k, s, t, $\lambda$ are parameters of OA. $\lambda$ can be derived from other parameters ($\lambda = N/s^t$). So an OA is represented as OA(N, k, s, t).

An example of OA(8, 4, 2, 3) is given below.

0 0 0 0
0 0 1 1
0 1 0 1
0 1 1 0
1 0 0 1
1 0 1 0
1 1 0 0
1 1 1 1

## 3. **Metrics of key predistribution Scheme**

Metrics to judge a given key predistribution scheme are: *connectivity*, *resiliency*, *network size* and *scalability*, *computation and communication overhead*, *complexity of shared-key discovery* and *path-key establishment*. We illustrate few parameters using the example given below.

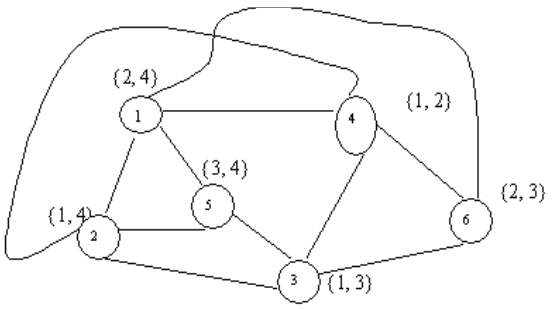


Figure 1. A sensor network of six sensor nodes

There are six sensor nodes in the sensor network shown in the above figure. The key pool is {1, 2, 3, 4}. Each sensor has two keys. For example, node 1 has the key-chain {2, 4}, node 2 has the key-chain {1, 4}, and so on. We assume all these sensors are in the communication range of each other. So any two nodes in this network can communicate if they have a common key. For example, Node 3 and Node 4 share a common key 1. So there is a communication link between them. But nodes 4 and 5 do not share any key, so there is no communication link between them. Given any two nodes, *connectivity* or *connection probability* defines the probability that there is a common key between them. In the above example, there are 6 nodes. So maximum number of connections possible in the network is $\binom{6}{2}$ = 15. The number of direct communication links that exist in the network is 11. So connection probability p = $\frac{11}{15}$ = 0.733. There are two kinds of *resiliency*: E(s), and V(s). Given a sensor network, if s nodes are compromised, then
E(s) = number of communication links broken / total number of communication links,
V(s) = number of victim nodes / total number of nodes.
In our example, let us suppose that s = 2 and attacker has captured the nodes 1 and 2. So she is able to extract the keys used by these nodes. This compromised key pool set is {1, 2, 4}. As a result we can check that among eleven communication links total nine communication links are compromised. So E(2) = $\frac{9}{11}$ = 0.818. And we see that due to this compromise of nodes 1 and 2, a different node that is node 4 is no more able to communicate with no other node. Because all keys of this node are compromised. In other words this node 4 has become victim node due to the compromise of nodes 1 and 2. Note that no other node turns victim node. So here V(2) = $\frac{1}{15}$ = 0.066. Scalability means given the same key pool and same key-chain length the ability to add new nodes in the distributed sensor network and assign them key-chains.

## 4. Orthogonal Array based KPS

J Dong et al proposed key predistribution scheme based on orthogonal array design in [12]. They have used OA of index one. Further they have used *Bush's construction* for constructing OA of index one.

## 4.1 Bush's construction method of OA of index unity

Step 1: Start.

Step 2: GF(s) is a Galois Field with $s=q^n$ elements. These elements are denoted by $e_i$ for i=0, 1, … , s-1.

Step 3: Consider the polynomial

$y_j(x)=a_{t-1}*x^{t-1} + a_{t-2}*x^{t-2} + \ldots + a_1*x + a_0$, where $a_i \in GF(s)$

Step 4: So we can have $s^t$ polynomials (j=0,1,…,$s^t - 1$).

Step 5: Form an s by $s^t$ array by inserting u at OA[i, j] such that

$y_j(e_i)= e_u \bmod q$.

Step 6: Stop.

## 4.2 Example of Bush's Construction

Let $q^n=5$ and t=2.. Note that q always has to be a prime number. So $q^n$ is a prime power, where n is a positive integer. So we get s=5, and $e_0=0$, $e_1=1$, $e_2=2$, $e_3=3$, $e_4=4$, and the polynomial is $y_j(x)= a_1*x + a_0$ , where $a_i \in GF(5)$. So there are 25 polynomials such as

$y_0(x)= 0*x + 0$
$y_1(x)= 0*x + 1$
$y_2(x)= 0*x + 2$
$y_3(x)= 0*x + 3$
$y_4(x)= 0*x + 4$
$y_5(x)= 1*x + 0$
$y_6(x)= 1*x + 1$
$y_7(x)= 1*x + 2$
$y_8(x)= 1*x + 3$
$y_9(x)= 1*x + 4$
$y_{10}(x)= 2*x + 0$ , and so on.

## 4.3 Construction of the orthogonal Array

Suppose we take a polynomial $y_8(x)= x +3$

$y_8(0)= 3$
$y_8(1)= 4$
$y_8(2)= 5 \bmod 5 =0$
$y_8(3)= 6 \bmod 5 = 1$
$y_8(4)= 7 \bmod 5 = 2$

so we get the $8^{th}$ row of array: 3 4 0 1 2 y
this y i.e., OA[i,q+1] term is filled up as the coefficient of the leading term. So here y=1.

### 4.4 **The Key Predistribution Scheme**

This OA is used to construct a combinatorial design (X, B ) with v = |X| = q*(q+1) = $q^2$+1, b = $q^t$ , k = q+1. That is, Size of key pool = $q^2$+1, number of sensor nodes = $q^t$, number of keys in each node = q+1. Elements from different columns of OA are considered as different elements of OA. So, Key Pool, X = { $a_{i,j}$ | 0≤ i ≤q-1, 0≤ j ≤q }. For example, we take the same example as above, i.e. $q^n$=5, t=2. The resultant orthogonal array following Bush's construction is shown in the table below.

| Column 0 | Column 1 | Column 2 | Column 3 | Column 4 | Column 5 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 | 4 | 4 |
| 0 | 1 | 2 | 3 | 4 | 1 |
| 1 | 2 | 3 | 4 | 0 | 1 |
| 2 | 3 | 4 | 0 | 1 | 1 |
| 3 | 4 | 0 | 1 | 2 | 1 |
| 4 | 0 | 1 | 2 | 3 | 1 |
| 0 | 2 | 4 | 1 | 3 | 2 |
| 1 | 3 | 0 | 2 | 4 | 2 |
| 2 | 4 | 1 | 3 | 0 | 2 |
| 3 | 0 | 2 | 4 | 1 | 2 |
| 4 | 1 | 3 | 0 | 2 | 2 |
| 0 | 3 | 1 | 4 | 2 | 3 |
| 1 | 4 | 2 | 0 | 3 | 3 |
| 2 | 0 | 3 | 1 | 4 | 3 |
| 3 | 1 | 4 | 2 | 0 | 3 |
| 4 | 2 | 0 | 3 | 1 | 3 |
| 0 | 4 | 3 | 2 | 1 | 4 |
| 1 | 0 | 4 | 3 | 2 | 4 |
| 2 | 1 | 0 | 4 | 3 | 4 |
| 3 | 2 | 1 | 0 | 4 | 4 |
| 4 | 3 | 2 | 1 | 0 | 4 |

Table 1. An orthogonal array obtained following Bush's construction

In the KPS, the elements from different columns of OA are considered as different elements of OA. That is, in the above example, elements of the first row would become (0,0), (0,1), (0,2), (0,3), (0,4), (0,5). So we get a two-dimensional array of elements of the form (x,y). Each of these elements is a key identifier. Some mapping technique is used to transform this key identifier to a cryptographic key of sufficient bit-length. In KPS literature, this key-identifier is loosely referred as key. Here key pool is the set {(x,y) | 0≤ x ≤4, and 0≤ y ≤5}. Each row represents the key-chain that is assigned

to a particular sensor node. So the resultant key predistribution is shown in the following table.

| Sensor Node | Key-chain | | | | | |
|---|---|---|---|---|---|---|
| | $Key_0$ | $Key_1$ | $Key_2$ | $Key_3$ | $Key_4$ | $Key_5$ |
| 0 | (0,0) | (0,1) | (0,2) | (0,3) | (0,4) | (0,5) |
| 1 | (1,0) | (1,1) | (1,2) | (1,3) | (1,4) | (1,5) |
| 2 | (2,0) | (2,1) | (2,2) | (2,3) | (2,4) | (2,5) |
| 3 | (3,0) | (3,1) | (3,2) | (3,3) | (3,4) | (3,5) |
| 4 | (4,0) | (4,1) | (4,2) | (4,3) | (4,4) | (4,5) |
| 5 | (0,0) | (1,1) | (2,2) | (3,3) | (4,4) | (1,5) |
| 6 | (1,0) | (2,1) | (3,2) | (4,3) | (0,4) | (1,5) |
| 7 | (2,0) | (3,1) | (4,2) | (0,3) | (1,4) | (1,5) |
| 8 | (3,0) | (4,1) | (0,2) | (1,3) | (2,4) | (1,5) |
| 9 | (4,0) | (0,1) | (1,2) | (2,3) | (3,4) | (1,5) |
| 10 | (0,0) | (2,1) | (4,2) | (1,3) | (3,4) | (2,5) |
| 11 | (1,0) | (3,1) | (0,2) | (2,3) | (4,4) | (2,5) |
| 12 | (2,0) | (4,1) | (1,2) | (3,3) | (0,4) | (2,5) |
| 13 | (3,0) | (0,1) | (2,2) | (4,3) | (1,4) | (2,5) |
| 14 | (4,0) | (1,1) | (3,2) | (0,3) | (2,4) | (2,5) |
| 15 | (0,0) | (3,1) | (1,2) | (4,3) | (2,4) | (3,5) |
| 16 | (1,0) | (4,1) | (2,2) | (0,3) | (3,4) | (3,5) |
| 17 | (2,0) | (0,1) | (3,2) | (1,3) | (4,4) | (3,5) |
| 18 | (3,0) | (1,1) | (4,2) | (2,3) | (0,4) | (3,5) |
| 19 | (4,0) | (2,1) | (0,2) | (3,3) | (1,4) | (3,5) |
| 20 | (0,0) | (4,1) | (3,2) | (2,3) | (1,4) | (4,5) |
| 21 | (1,0) | (0,1) | (4,2) | (3,3) | (2,4) | (4,5) |
| 22 | (2,0) | (1,1) | (0,2) | (4,3) | (3,4) | (4,5) |
| 23 | (3,0) | (2,1) | (1,2) | (0,3) | (4,4) | (4,5) |
| 24 | (4,0) | (3,1) | (2,2) | (1,3) | (0,4) | (4,5) |

Table 2. Key predistribution following the D. Pei et el scheme.

## 5. Analysis of OA based KPS

In [12], authors have compared their scheme with other existing KPS based on two parameters:

(1) Connection Probability
(2) Probability fail(1)

The parameter fail(1) denotes the probability of a node becomes victim node when a single node is compromised. In the following table we present the connection probability and resiliency of this scheme.

| Value of q | Value of t | N | Connection Probability, P | E(1) | V(1) |
|---|---|---|---|---|---|
| 5 | 3 | 125 | 0.707 | 0.218 | 0.707 |
| 5 | 4 | 625 | 0.757 | 0.216 | 0.757 |
| 7 | 3 | 343 | 0.648 | 0.963 | 0.733 |
| 7 | 4 | 2401 | 0.726 | 0.148 | 0.0004 |
| 7 | 5 | 16807 | 0.715 | 0.148 | 0.00006 |
| 11 | 2 | 121 | 0.992 | 0.0916 | 0.00826 |
| 11 | 3 | 1331 | 0.593 | 0.092 | 0.0007 |
| 11 | 4 | 14641 | 0.701 | 0.092 | 0.00006 |
| 13 | 2 | 169 | 0.994 | 0.774 | 0.0059 |
| 13 | 3 | 2197 | 0.579 | 0.077 | 0.0004 |
| 13 | 4 | 28561 | 0.695 | 0.0776 | 0.000035 |

Table 3. Measurement of Connection Probability, and of Resiliency when 1 node is compromised in Dingyi Pei et el scheme.

## 6. Weakness of the Scheme

Upon simulation of this scheme we observe very serious flaw in the scheme. When we compromise 5% or 10% of the total number of sensor nodes, which is very natural, we see that resiliency becomes very poor. Both the E(s) and V(s) becomes close to 1.0. That means all the connections are getting compromised if we compromise only 5% or 10% of nodes. Also almost all the remaining sensor nodes are becoming victim nodes in the sense they can no longer communicate with other nodes as their all keys get compromised. In the following table we summarize these observations.

| Value of q | Value of t | N | Connection probability $p_1$ | Number of compromised nodes, s | E(s) | V(s) |
|---|---|---|---|---|---|---|
| 5 | 2 | 25 | 0.966 | 3 | 0.554 | 0.12 |
| 5 | 2 | 25 | 0.966 | 5 | 0.698 | 0.20 |
| 5 | 3 | 125 | 0.707 | 7 | 0.798 | 0.21 |
| 5 | 3 | 125 | 0.707 | 13 | 0.982 | 0.872 |
| 7 | 3 | 343 | 0.648 | 17 | 0.963 | 0.733 |
| 7 | 3 | 343 | 0.648 | 34 | 1.0 | 0.998 |
| 7 | 4 | 2401 | 0.726 | 120 | 1.0 | 0.999 |
| 7 | 4 | 2401 | 0.726 | 240 | 1.0 | 0.999 |

Table 4. Resiliency of the KPS when 5% and 10% nodes are compromised

Since number of nodes, N = q*t, we can get larger N by either increasing q or by increasing t or by increasing both. In the above simulation we have taken prime power q small . For similar values of N, we wanted to study if there is any change in the performance of the scheme if we take prime power q larger and keep t small. We observe that the scenario does not change much. The result of simulation is shown below.

| Value of q | Value of t | N | Connection probability $p_1$ | Number of compromised nodes, s | E(s) | V(s) |
|---|---|---|---|---|---|---|
| 11 | 2 | 121 | 0.992 | 6 | 0.442 | 0.049 |
| 11 | 2 | 121 | 0.992 | 12 | 0.687 | 0.099 |
| 19 | 2 | 361 | 0.997 | 18 | 0.632 | 0.049 |
| 19 | 2 | 361 | 0.997 | 36 | 0.863 | 0.111 |
| 47 | 2 | 2209 | 0.999 | 110 | 0.913 | 0.054 |
| 47 | 2 | 2209 | 0.999 | 220 | 0.993 | 0.745 |

Table 5. Resiliency of the KPS when 5% and 10% nodes are compromised with changed parameters for similar N as of Table 4.

The reason for such poor resiliency is the small key pool size. This small key pool size is also the reason of good connectivity of this OA based KPS. This key pre-distribution scheme is suitable only for sensor network of very small size.

## 7. Future Work

In the work by J Dong et al, they have used Bush's construction for the construction of orthogonal array. One can use different orthogonal array construction schemes and look if that gives good results.

## 8. Conclusion

*Designs provide balanced set systems*. This balance yields some good algorithmic consequences. One consequence is that it helps us to get efficient description of key establishment algorithms when designs are used for key predistribution. Key predistribution using combinatorial designs are an area of active research. *This work will definitely help researchers to* get detail understanding of the key predistribution scheme using orthogonal array done by J Dong et al and its weakness in terms of resiliency. One can further look for different orthogonal array construction schemes for proposing improved key predistribution scheme.

## References


[1] C. J. Colbourn & P. C. Van Oorschot, Applications of Combinatorial Design in Computer Science, ACM Computing Surveys, Vol. 21, No. 2, June 1989.

[2] C. J. Colbourn, J. H. Dinitz, D. R. Stinson, Application of Combinatorial Design in Communications, Cryptography, and Networking,

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A Survey on Sensor Networks,

[4] Praveen Rentala, Ravi Musunuri, Shashidhar Gandham, Udit Saxena, Survey On Sensor Networks, Proceedings of International Conference on Mobile Computing and Networking, 2001.

[5] Marcos A. M. Vieira, Claudinor N. Coelho. Jr., DiÓgenes C. Da Silva Jr., José M. da Mata, Survey on Wireless Sensor Network Devices, Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03, IEEE Conference**.**

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless Sensor Networks: a Survey, Computer Networks 38 (2002) 393-422.

[7] S. A. Camtepe & B. Yener, Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks, IEEE/ACM Transactions on Networking, Vol. 15, No. 2, April 2007.

[8] S. A. Camtepe & B. Yener, Key Distribution Mechanisms for Wireless Sensor Netowork: a Survey, Rensselaer Polytechnic Institute Technical Report TR-05-07 (March 2005).

[9] J. Lee & D. R. Stinson, A Combinatorial Approach to Key Predistribution for Distributed Sensor Networks, IEEE Communications Society/WCNC 2005.

[10] Sushmita Ruj & Bimal Roy, Key Predistribution using PBIBD in Wireless Sensor Network, ISPA 2007, LNCS 4742, pp. 431-445, 2007.

[11] Sushmita Ruj & Bimal Roy, Key Establishment Algorithms for some Deterministic Key Predistribution Schemes, WOSIS 2008: 68-77

[12] Junwu Dong, Dingyi Pei, Xueli Wang, A Class of Key Predistribution Based on Orthogonal Arrays, Journal of Computer Science and Technology 23(5):825-831 Sept. 2008.

[13] Junwu Dong, Dingyi Pei, Xueli Wang, A Class of Key Predistribution Based on 3-Designs, Inscrypt 2007, LNCS 4990, pp.81-92, 2008.

[14] Street A. P. and Street D. J, Combinatorics of Experimental Design, Clarendon Press, Oxford (1987).

[15] D. R. Stinson, Combinatorial Designs: Constructions and Analysis. Springer-Verlag, New York (2004).