2009

Key predistribution schemes using block designs in wireless sensor networks

Sushmita Ruj
Indian Statistical Institute, sush_r@isical.ac.in

J. Seberry
University of Wollongong, jennie@uow.edu.au

Bimal Roy
Indian Statistical Institute, bimal@isical.ac.in

Key predistribution schemes using block designs in wireless sensor networks

Abstract

We present a deterministic key predistribution scheme in wireless sensor networks (WSN) using combinatorial designs. The design finds application where a large number sensor nodes are to be deployed. This scheme has the advantage that each sensor node contains very few keys, however every pair of sensor nodes within communication range can directly communicate with each other. We calculate the resiliency of the network with respect to two parameters of resiliency. Our scheme is resilient to selective node capture attack and node fabrication attack.

Disciplines

Physical Sciences and Mathematics

Publication Details

Ruj, S., Seberry, J. & Roy, B. (2009). Key predistribution schemes using block designs in wireless sensor networks. 12th IEEE International Conference on Computational Science and Engineering (CSE 2009) (pp. 873-878). USA: IEEE.

Key Predistribution Schemes Using Block Designs in Wireless Sensor Networks

Sushmita Ruj
Applied Statistics Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, INDIA
sush_r@isical.ac.in

Jennifer Seberry
School of Information Technology and Computer Science,
University of Wollongong,
Wollongong, NSW, 2522, AUSTRALIA
jennie@uow.edu.au

Bimal Roy
Applied Statistics Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, INDIA
bimal@isical.ac.in

Abstract

We present a deterministic Key Predistribution scheme in Wireless Sensor Networks (WSN) using combinatorial designs. The design finds application where a large number sensor nodes are to be deployed. This scheme has the advantage that each sensor node contains very few keys, however every pair of sensor nodes within communication range can directly communicate with each other. We calculate the resiliency of the network with respect to two parameters of resiliency. Our scheme is resilient to selective node capture attack and node fabrication attack.

Keywords: Combinatorial Design, Intersection Number, SBIBD.

1 Introduction

Sensor nodes are devices with very limited power and memory and are deployed in large numbers over a target region. There are two types of sensor networks. (i)Hierarchical Wireless Sensor Networks (HWSN) and (ii)Distributed Wireless Sensor Networks (DWSN). HWSN consists of sensor nodes with different power and memory and are deployed according to some predetermined pattern. There are mainly three types of nodes according to descending capabilities. (i) Base stations (ii) Cluster heads (iii) Sensor nodes. The DWSN contain sensor nodes which are randomly deployed and continue to work unattended for large periods of time. Sensor nodes can effectively communicate with each other within a particular range called the Radio Frequency (RF) range.

In applications where secure communication in needed (for example in adversarial regions), sensors encrypt messages using cryptographic keys. The keys are either predistributed in the sensor nodes or online key exchange protocols can be used. Online key exchange is not very popular to date as implementation of public key framework demands processing power at the higher end. Very recently implementations of ECC and RSA on 8-bit CPUs have been proposed [13]. Still a closer scrutiny of [12, Table 2, Section 3. 3] reveals that the algorithms execute in seconds (the range being 0.43s to 83.26s); whereas the key predistribution just involves the calculation of inverse of an integer modulo a prime number [14], which is bound to be much faster than the former. Hence key predistribution is an attractive option.

Key predistribution techniques can be randomized, deterministic or hybrid. In randomized technique of key predistribution [11, 7], keys are drawn randomly from a key pool and placed in each sensor node. This technique does not guarantee that any two nodes will be able to communicate directly. If direct communication is not possible, then a path needs to be established between the two nodes. This makes communication slower and power consuming.

In deterministic key predistribution, keys are placed in sensor nodes in a predetermined manner. The pioneering

978-0-7695-3823-5/09 $26.00 © 2009 IEEE
DOI 10.1109/CSE.2009.35

work of Camtepe and Yener [3, 4] used projective planes and generalized quadrangles, Lee and Stinson [14, 15] used transversal designs, Chakrabarty, Maitra and Roy [5, 6] used merging blocks constructed from transversal designs. Other works include the use of $PBIBD$ by Ruj and Roy [18], modified transversal designs by Ruj and Roy [17], $3-designs$ by Dong, Pei and Wang [9] and *Costas arrays* and Distinct Difference Configuration (DDC) by Blackburn, Etzion, Martin and Paterson [1, 2] and orthogonal arrays [10]. Hybrid designs combine the above two approaches and have been studied in [3, 4, 5].

Here we consider a deterministic key predistribution scheme based on combinatorial designs. It has the advantage that it can support a network of very large size, at the cost of very few keys in each node. Also by suitably choosing the parameters of the design, it can be ensured that every pair of nodes within communication range can communicate directly, thus making communication efficient and less error-prone. The main advantage of our scheme is that it is resilient to selective node capture attack and node fabrication attack.

The rest of the paper is organized as follows. In Section 2 we define a few terms and concepts. We also define the threat model. We present our predistribution scheme in Section 3.

We study the effect of node compromise on such a network in Section 4. We conclude with some open problems in Section 5.

2 Preliminaries

2.1 Combinatorial designs

Definition 1 *A **set system** or **design** [14] is a pair (X, A), where A is a set of subsets of X, called **blocks**. The elements of X are called **varieties**. A Balanced Incomplete Block Design (BIBD) $BIBD(v, b, r, k, \lambda)$, is a **design** which satisfy the following conditions:*

1. *$|X| = v$, $|A| = b$,*
2. *Each subset in A contains exactly k elements,*
3. *Each variety in X occurs in r many blocks,*
4. *Each pair of varieties in X is contained in exactly λ blocks in A.*

A $BIBD(v, b, r, k, \lambda)$ design can be represented by a *incidence matrix* $M = [m_{ij}]$ of dimension $v \times b$ with entries 0 and 1. $m_{ij} = 1$, if the ith variety is present in the jth block and 0 otherwise.

Definition 2 *Suppose that (X, A) is a set system, where*

$$X = \{x_i : 1 \leq i \leq v\}$$

and

$$A = \{A_j : 1 \leq j \leq b\}.$$

*The **dual set system** of (X, A) is any set isomorphic to the set system (X', A') where*

$$X' = \{x'_j : 1 \leq j \leq b\},$$

$$A' = \{A'_i : 1 \leq i \leq v\},$$

and where

$$x'_j \in A'_i \iff x_i \in A_j.$$

It follows that if we take the dual of a $BIBD(v, b, r, k, \lambda)$, we arrive at a design containing b varieties, v blocks each block containing exactly r varieties and each variety occurring in exactly k blocks. We also note that any two blocks contain λ elements in common.

Definition 3 *When $v = b$, the $BIBD$ is called a **symmetric BIBD** or **SBIBD** and denoted by $SB[v, k; \lambda]$.*

Definition 4 *A **pairwise balanced design (PBD)** is a design in which each pair of points occurs in λ blocks, for some constant λ, called the **index** of the design.*

Definition 5 *The **intersection number** between any two blocks is the number of elements common to the blocks.*

Definition 6 *Let the intersection numbers between any the blocks in a $BIBD$ be μ_1, μ_2, $\cdots$, μ_x. Let $M = \{\mu_i : i = 1, 2, \cdots, x\}$. Let $\mu = max\{\mu_1, \mu_2, \cdots, \mu_x\}$. μ is called the **linkage** of the design.*

We note that for a $SBIBD$, $|M| = 1$ and $\mu = \lambda$.

2.2 Threat Model

There are different types of models for node capture [16].

1. **Random node capture attack:** Nodes are captured randomly.
2. **Selective node capture attack:** This capture attack is given in [16]. Assume that the attacker's goal is to collect a subset T of the keys in the pool. The attacker has already compromised a number of sensors, and has collected all their keys in a set W. For every sensor s in the WSN, the *key information gain* $G(s)$ is a random variable equal to the number of keys in the key ring of s which are in T and are not in W. For example, if the attacker's goal is to compromise the channel

between sensors s_a and s_b, subset T in the above definition is equal to $M_a \bigcap M_b$, (where M_a and M_b are the keys in the key chains of s_a and s_b respectively) that is it contains all the keys which are in the key-ring of both s_a and s_b . Assuming that the attacker has collected a set W of keys, random variable $G(s)$ is equal to $|(M_s \bigcap M_a \bigcap M_b) \backslash W|$. At each step of the attack sequence, the next sensor to be tampered with is sensor s, where s maximizes $E[G(s)|I(s)]$, the expectation of the key information gain $G(s)$ given the information $I(s)$ that the attacker knows on sensor s key ring.

3. **Node Fabrication Attack:** In the node fabrication attack, the attacker compromises only a few sensors and uses the captured keys to fabricate sensors with identities of uncompromised sensors or fabricate sensors with new identities. Then, the attacker can deploy the fabricated nodes in the parts of the network where the original node is not present. The uncompromised sensors in the network cannot detect the fabricated nodes as anomalous nodes as long as they can have standard communication with them. This attack is more severe as compared to passive listening attacks as the attacker may have enough information to fabricate many sensors with many different identities and possibly outnumber the original set of sensors.

In this paper we show that an attacker does not gain in any way by launching a selective node capture attack. In fact selective node capture is just as good as random node capture from the point of view of the attacker. We show that our scheme is resilient to node fabrication attack.

3 Design construction

We can map a $BIBD(v, b, r, k, \lambda)$ design to a sensor network containing v keys in the key-pool. There are b sensor nodes, each node containing k keys are each key occurring in r nodes. Any pair of keys occur in λ blocks.

It can be seen that for a symmetric design any pair two blocks will contain λ elements in common, since $\mu = \lambda$.

We consider two BIBDs: $D_1 = (v_1, b_1, r_1, k_1, \lambda_1)$ and $D_2 = (v_2, b_2, r_2, k_2, \lambda_2)$. Let $M_1 = [m'_{ij}]$ and $M_2 = [m''_{ij}]$ be the respective incidence matrices. Therefore the dimension of M_1 and M_2 are $v_1 \times b_1$ and $v_2 \times b_2$ respectively. A requirement for our design is that $k_1 = v_2$. This facilitates in the construction of the new matrix from the older ones. We construct the matrix M in the following way.

1. For every column j of M_1 replace m'_{ij} by a row of M_2, if $m'_{ij} = 1$. For each i replace $m'ij$ by a different row of M_2.
2. For every column j of M_1 replace m'_{ij} by a row vector of length b_2 containing all zeros, if $m'_{ij} = 0$.

The result of the following operations is a matrix M of dimension $v_1 \times b_1 b_2$. We call the design D and represent D by $D = D_1 \bowtie D_2$. We say that D is the *expanded design* of D_1 and D_2. The blocks in D which arise from a given block in D_1 are said to belong to the same group.

Now we map this design D to a sensor network consisting of $b_1 b_2$ nodes, each node consisting of k_2 keys. The key-pool consists of v_1 keys.

3.1 Analysis of the linked design

We note that the number of sensor nodes can be increased without increasing the size of the key-pool and the number of keys in each node. This is very important since a $DWSN$ contains a large number of nodes with very limited memory and power. Both these problems can be effectively handled by our design.

Now we have to ensure that no two nodes will have the same set of keys in them. The following example results in two nodes having the same set of keys.

Example 1 *Consider the designs $D_1 = (8, 14, 7, 4, 3)$ and $D_2 = (4, 4, 3, 3, 2)$ [8]. Then we arrive at the matrices which are given below.*

$$D_1 = \begin{matrix} 1&1&1&1&1&1&1&0&0&0&0&0&0&0\\ 1&1&1&0&0&0&0&1&1&1&1&0&0&0\\ 1&1&0&1&0&0&0&1&0&0&0&1&1&1\\ 1&0&0&0&1&1&0&0&1&1&0&1&1&0\\ 0&1&0&0&1&1&1&0&1&0&1&1&0&1\\ 0&0&1&1&1&1&0&0&0&1&1&0&1&1\\ 0&0&1&0&0&0&1&1&1&0&0&0&1&1\\ 0&0&0&1&0&0&1&1&0&1&1&1&0&0 \end{matrix}$$

$$\text{and } D_2 = \begin{matrix} 1&1&1&0\\ 1&1&0&1\\ 1&0&1&1\\ 0&1&1&1 \end{matrix}.$$

For columns say 1 and 2 of D_1, we get two identical columns in in $D_1 \bowtie D_2$ as

$$\begin{matrix} 1&1&1&0\\ 1&1&0&1\\ 1&0&1&1\\ 0&1&1&1\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{matrix} \text{ and } \begin{matrix} 1&1&1&0\\ 1&1&0&1\\ 1&0&1&1\\ 0&0&0&0\\ 0&1&1&1\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{matrix}.$$

So we see that two blocks (corresponding to columns 1 and 5 in $D_1 \bowtie D_2$) will be identical.

The following theorem imposes some restriction on the choices of the parameters of the design so that no two blocks are identical.

Theorem 1 *Let* $D_1 = (v_1, b_1, r_1, k_1, \lambda_1)$ *and* $D_2 = (v_2, b_2, r_2, k_2, \lambda_2)$ *be two* $BIBDs$. *Let* μ_1, μ_2, $\cdots$, μ_t *be the intersection numbers of any two blocks in* D_1. *Let* $\mu = max\{\mu_i : i = 1, 2, \cdots, t\}$. *If* $\mu < k_2$ *then, no two blocks will have the same set of keys in them.*

Proof : Refer to the construction in Section 3. By our construction each column gives rise to b_2 columns. The b_2 blocks corresponding to these columns which belong to the same group will be different since the blocks in D_2 are different. So we consider blocks which arise out of two different columns of D_1. Let B_1 and B_2 be two blocks in D_1 which share μ elements in common. So the matrix M_1 will have μ rows where both the columns B_1 and B_2 will have ones in them. Let these rows be n_1, n_2, $\cdots$, n_μ. When we construct the matrix M, for each of these k_1 ones of column B_1 and B_2 we substitute a row of M_2. Let for the μ rows n_1, n_2, $\cdots$, n_μ we have ones in the same rows in two columns C_1 arising from B_1 and column C_2 arising from B_2 of the matrix M. Since $\mu < k_2$ there are some other rows in the block B_1 which will give a 1 in column C_1 but not in C_2. Similarly, there are some other rows in the block B_2 which will give a 1 in column C_2 but not in C_1. Hence the two nodes arising from C_1 and C_2 can never have the same keys. This happens for every pair of columns in the resulting matrix M. Hence none of the blocks in M have the same elements. ■

Additionally we would like to ensure that any two blocks share at least one key. We give a construction which will always ensure that any two blocks will share at least one key. Let $D_1 = (v_1, b_1, r_1, k_1, \lambda_1)$ such that two blocks share μ_1, μ_2, $\cdots$, μ_i varieties in common. Let $\mu = min\{\mu_i : i = 1, 2, \cdots, t\} \geq 3$. Let $D_2 = (v_2 = k_1, b_2 = k_1, r_2 = k_1 - 1, k_2 = k_1 - 1, \lambda_2 = k_1 - 2)$. Then it can be seen that any two blocks will have at least one element in common.

Example 2 *Consider the two designs* $D_1 = (63, 63, 31, 31, 15)$ *and* $D_2 = (31, 31, 30, 30, 29)$, *then the sensor network will have* 1953 *sensors each sensor having just* 30 *keys and the size of the key-pool will be* 63 *and each pair of nodes can communicate directly with each other.*

Example 3 *Consider the two designs* $D_1 = (255, 255, 127, 127, 63)$ *and* $D_2 = (127, 127, 126, 126, 125)$, *then the sensor network will have* 32385 *sensors each sensor having just* 126 *keys which is much less than the square root of the number of nodes. The size of the key-pool will be* 255 *and each pair of nodes shares more than one keys.*

The above example are just a few instances to show how we can increase the number of sensor nodes, without adding more keys in the nodes and communicate directly with every node thus saving power and minimizing errors caused due to multiple hops.

We next study the effect of node compromise. We define two parameters for resiliency and and show that our scheme fares better than that of [14] in several respects.

4 Effect of node compromise

Sensor nodes deployed in an hostile region are prone to node capture or compromise. In such a situation all the keys in the compromised nodes become ineffective and cannot be used for further communication. Hence we need to know how resilient the network is under node compromise. This means that on compromising some nodes only a part of the network will be affected. When nodes are compromised, it may so happen that either some links are broken or a whole node is disconnected. The later happens when all the keys in the node are exposed. We measure the resiliency of a network in terms of two parameters $V(s)$, which is the fraction of nodes disconnected when s nodes are compromised and $E(s)$, which is the fraction of links broken when s nodes are compromised.

We have already seen that there can be three types of attack on the sensor nodes.

4.1 Resiliency against selective node capture

During selective node capture attack, the attacker compromises those nodes whose keys have not already been compromised. We note that any two nodes broadcast only their node identifiers during the shared-key discovery phase. The key identifiers are not broadcasted. At no stage the attacker can know what key identifier is present in which node. Hence there is no way of knowing which nodes are left to be compromised. Thus unless the attacker compromises the node, she cannot choose a node for compromise to maximize the number of keys compromised. Hence the attacker does not gain anything by mounting a selective node capture.

4.2 Resiliency against node fabrication attack

In node fabrication attack, the attacker compromises a few sensors and fabricates new nodes with new identities or with identities of the uncompromised sensors. In our scheme each node has a distinct identifier and hence it is not possible to assign the same identifier to another node. Also since the nodes know which identifiers are valid, new

N	Size of Key-pool	No of keys per node	s	$V(s)$
528	66	23	6	0.1667
1596	288	27	8	0.0681
1840	184	39	7	0.12711
2840	355	39	8	0.08264
2840	355	39	10	0.09641

Table 1. Experimental value of $V(s)$ for 100 runs, when number of nodes is N and s nodes are compromised.

identifiers cannot be assigned. Hence our scheme is secure against node fabrication attack.

4.3 Resiliency against random node compromise

Study of $V(s)$:

$V(s)$ is defined as the fraction of nodes disconnected when s nodes are compromised. Mathematically,

$$V(s) = \frac{\text{Number of nodes disconnected when } s \text{ nodes are compromised}}{N-s},$$

where N is the size of the network.

We calculate the minimum number of nodes that must be compromised to disconnect one node. To disconnect one node all the k_2 keys in the node must be compromised. We note that if nodes within the same group are compromised, maximum nodes are affected. This is because, two nodes within the same group intersect at more number of points that two nodes in distinct blocks. When the parameters are $(v_2, v_2, v_2-1, v_2-1, v_2-2)$, then any two nodes within the same group share v_2-2 keys. Thus if two nodes are compromised within the same block, then one node is disconnected. This node lies in the same block as the compromised nodes.

However if nodes belong to different blocks then more number of nodes have to be compromised to disconnect one node.

The Table 1 gives the experimental result of $V(s)$ when s nodes are compromised.

Study of $E(s)$

To break the entire network, the minimum number of nodes that have to be compromised is v_1/k_2. This is because to break the entire system all keys v_1 have to be exposed. Suppose each sensor contributes k_2 keys. Given s nodes are compromised ($s < v_1/k_2$), number of keys lost is less than sk_2. Number of links broken is total number of links $* sk_2/v_1$. Hence maximum value of $E(s)$ is given by sk_2/v_1. $E(s)$ denotes the fraction of links broken when s nodes are compromised. Mathematically,

$$E(s) = \frac{\text{Number of links broken when } s \text{ nodes are compromised}}{(N-s)(N-s-1)/2},$$

where s is the number of nodes compromised. The Table 2 represent the experimental values of $E(s)$.

N	Size of Key-pool	No of keys per node	s	$E(s)$
528	66	23	7	0.4712
1596	288	27	8	0.3807
1840	184	39	7	0.3485
2840	355	39	8	0.2547
2840	355	39	10	0.3443

Table 2. Experimental value of $E(s)$ for 100 runs, when number of nodes is N and s nodes are compromised.

4.4 Comparative study

We compare our design with that given in [14] and see that our design performs much better in several respects. Firstly even for a very large network the number of keys per node is very small. If N be the size of the network, then each node contains less than $\sqrt{N}$ keys. Secondly we can ensure that every pair of nodes within communication range is directly connected. This minimizes the cost, the time and the error in communication. The design given in [14] did not ensure that any two nodes were directly connected. Thirdly we see that our design has better resiliency ($E(s)$) as observed in Figure 1 below. We compare the following two schemes. We choose approximately the same parameters which are given in Lee and Stinson's [14] scheme.

1. Our scheme having a network of size 2840 where each node has 39 keys per node and the size of the key-pool is 355.

2. The scheme given in [14] having a network of size 2209 where each node has 30 keys and the size of the key-pool is 1740.

5 Conclusion

In this paper we describe a key predistribution scheme using combinatorial designs. Our scheme can support a very large network still maintaining very few keys in each

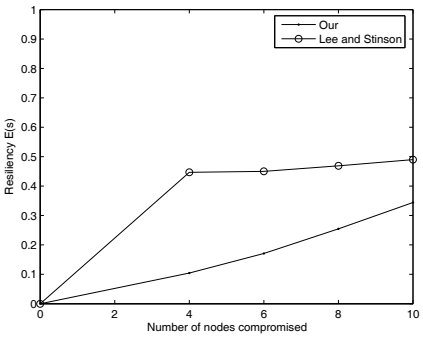

Figure 1. Resiliency ($E(s)$) Vs Number of nodes compromised in our scheme and Lee and Stinson's [14] scheme.

node. By properly choosing the parameters it can be ensured that any two nodes within communication range can communicate directly. Our scheme is also resilient to selective node capture attack and node fabrication attack.

In future we would like to study the properties of expanded design and chose the combinatorial designs which give best resiliency.

References

[1] S. R. Blackburn, T. Etzion, K. M. Martin, and M. B. Paterson. Efficient key predistribution for grid-based wireless sensor networks. In R. Safavi-Naini, editor, *ICITS*, volume 5155 of *Lecture Notes in Computer Science*, pages 54–69. Springer, 2008.

[2] S. R. Blackburn, T. Etzion, K. M. Martin, and M. B. Paterson. Key predistribution techniques for grid-based wireless sensor networks, 2009. Available at ePrint Cryptology archive 2009/014.

[3] S. A. Çamtepe and B. Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. In P. Samarati, P. Y. A. Ryan, D. Gollmann, and R. Molva, editors, *ESORICS*, volume 3193 of *Lecture Notes in Computer Science*, pages 293–308. Springer, 2004.

[4] S. A. Çamtepe and B. Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 15(2):346–358, 2007.

[5] D. Chakrabarti, S. Maitra, and B. K. Roy. A key predistribution scheme for wireless sensor networks: Merging blocks in combinatorial design. In J. Zhou, J. Lopez, R. H. Deng, and F. Bao, editors, *ISC*, volume 3650 of *Lecture Notes in Computer Science*, pages 89–103. Springer, 2005.

[6] D. Chakrabarti, S. Maitra, and B. K. Roy. A key predistribution scheme for wireless sensor networks: merging blocks in combinatorial design. *Int. J. Inf. Sec.*, 5(2):105–114, 2006.

[7] H. Chan, A. Perrig, and D. X. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, pages 197–213. IEEE Computer Society, 2003.

[8] C. J. Colbourn and J. H. Dinitz. *The CRC Handbook of Combinatorial Designs*. CRC Press, 1995.

[9] J. Dong, D. Pei, and X. Wang. A key predistribution scheme based on 3-designs. In D. Pei, M. Yung, D. Lin, and C. Wu, editors, *Inscrypt*, volume 4990 of *Lecture Notes in Computer Science*, pages 81–92. Springer, 2007.

[10] J. Dong, D. Pei, and X. Wang. A class of key predistribution schemes based on orthogonal arrays. *JCST*, 2008. To appear.

[11] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In V. Atluri, editor, *ACM Conference on Computer and Communications Security*, pages 41–47. ACM, 2002.

[12] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. In *MobiHoc*, pages 251–254. ACM, 2001.

[13] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In M. Joye and J.-J. Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 119–132. Springer, 2004.

[14] J. Lee and D. R. Stinson. A combinatorial approach to key predistribution for distributed sensor networks. In *IEEE Wireless Communications and Networking Conference, WCNC 2005, New Orleans, LA, USA*, 2005.

[15] J. Lee and D. R. Stinson. On the construction of practical key predistribution schemes for distributed sensor networks using combinatorial designs. *ACM Trans. Inf. Syst. Secur.*, 11(2), 2008.

[16] R. D. Pietro, L. V. Mancini, and A. Mei. Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks. *Wireless Networks*, 12(6):709–721, 2006.

[17] S. Ruj and B. Roy. Key predistribution using combinatorial designs for grid-group deployment scheme in wireless sensor networks. ACM Transaction on Sensor Networks. Accepted.

[18] S. Ruj and B. K. Roy. Key predistribution using partially balanced designs in wireless sensor networks. In I. Stojmenovic, R. K. Thulasiram, L. T. Yang, W. Jia, M. Guo, and R. F. de Mello, editors, *ISPA*, volume 4742 of *Lecture Notes in Computer Science*, pages 431–445. Springer, 2007.