Clique Size in Sensor Networks with Key Pre-distribution Based on Transversal Design*

DIBYENDU CHAKRABARTI, SUBHAMOY MAITRA and BIMAL ROY

Applied Statistics Unit, Indian Statistical Institute, Kolkata, India

Key pre-distribution is an important area of research in Distributed Sensor Networks (DSN). Two sensor nodes are considered connected for secure communication if they share one or more common secret key(s). It is important to analyse the largest subset of nodes in a DSN where each node is connected to every other node in that subset (i.e., the largest clique). This parameter (largest clique size) is important in terms of resiliency and capability towards efficient distributed computing in a DSN. In this paper, we concentrate on the schemes where the key pre-distribution strategies are based on transversal design and study the largest clique sizes. We show that merging of blocks to construct a node provides larger clique sizes than considering a block itself as a node in a transversal design.

Keywords Clique; Combinatorial Design; Distributed Sensor Networks; Key Pre-distribution; Random Merging; Secured Communication

1. Introduction

A sensor node is a small, inexpensive and resource constrained device that operates in RF (radio frequency) range. It has limitations in different aspects such as communication, computation, power, and storage. A DSN (distributed sensor network) is an ad-hoc network consisting of sensor nodes. The sensor nodes are often deployed in an uncontrolled environment where they are expected to operate unattended. In many situations, the DSN is also very large. In either case, though one might try to control the density of deployment, the only deployment option is to randomly scatter the nodes to cover the target area. The consequence is that the location or topology is not available prior to deployment.

Given the various limitations, the security of the DSN hinges on efficient key distribution techniques. Even with the present day technology, public key cryptosystems are considered too computation intensive for DSNs and typically a DSN establishes a secure network by the use of pre-distributed keys. The following four metrics are often used to evaluate key pre-distribution solutions.

1. Scalability: The distribution must allow post-deployment increase in the size of network.
2. Efficiency:
   (a) storage: Amount of memory required to store the keys.
   (b) computation: Number of cycles needed for key establishment

(c) communication: Number of messages exchanged during the key generation/ agreement phase.

3. Key Connectivity (probability of key share): The probability that two nodes share one/more keys should be high.
4. Resilience: Even if a number of nodes are compromised, i.e., the keys contained therein are revealed, the complete network should not fail, i.e., only a part of the network should be affected.

One of the challenges in DSNs is to find efficient algorithms to distribute the keys to sensor nodes before they are deployed. The solutions may be categorized as follows:

1. Probabilistic: The keys are randomly chosen from a given collection of keys and distributed to the sensor nodes.
2. Deterministic: The key distribution is obtained as the output of some deterministic algorithm.
3. Hybrid: A combination of deterministic and probabilistic approaches.

A trivial (and obvious) deterministic solution to the problem is to put the same key in all the nodes. However, the moment a single node is compromised, the network fails. To guard against such a possibility, one can think of using distinct keys for all possible pair of nodes in the DSN. The very good resilience notwithstanding, the solution is not viable for even networks of moderate size due to the limited storage capacity of the nodes. If there are N nodes, then there will be $\binom{N}{2}$ keys in total and each node must have $N-1$ many keys. It is not possible to accommodate $N-1$ many keys in a node given the current memory capacity of sensor hardware when N is moderately large, say ≥ 500.

Let us now briefly refer a few state of the art key pre-distribution schemes. The well known Blom's scheme [1] has been extended in recent works for key pre-distribution in wireless sensor networks [5, 7]. The problem with these kinds of schemes is the use of several multiplication operations (as example see [5, Section 5.2]) for key exchange. The randomized key pre-distribution is another strategy in this area [6]. However, the main motivation is to maintain a connectivity (possibly with several hops) in the network. As an example [6, Section 3.2], a sensor network with 10000 nodes has been considered and to maintain the connectivity, it has been calculated that it is enough if one node can communicate with only 20 other nodes. Note that the communication between any two nodes may require a large number of hops. However, only the connectivity criterion (with too many hops) may not suffice in an adversarial condition. Further in such a scenario, the key agreement between two nodes requires exchange of the key indices. The use of combinatorial and probabilistic design (also a combination of both – termed as hybrid design) in the context of key distribution has been proposed in [2]. In this case also, the main motivation was to have low number of common keys.

In [8] transversal design (see Subsection 2.1 for more details) has been used where the blocks correspond to the sensor nodes. In our recent works [3,4], we have proposed to start from a combinatorial design and then apply a probabilistic extension in the form of random merging of blocks to form the sensor nodes and in this case there is good flexibility in adjusting the number of common keys between any two nodes. In our earlier works [3,4], we dealt with the cases of

(i) unconstrained random merging of blocks and
(ii) random merging of blocks with the restriction that the nodes are composed of disjoint blocks (do not share common keys among themselves). The computation to find out a shared key under this framework is of very low time complexity [8, 3, 4], which

basically requires calculation of the inverse of an element in a finite field. That is the reason this kind of design becomes popular for application in key pre-distribution.

In the domain of distributed computing, the nodes forming a complete graph is an "ideal situation". As mentioned earlier, one gains a lot in terms of resilience. Moreover, the communication complexity decreases because fewer messages are exchanged between the nodes in order to generate/agree upon a key. In such a scenario, there is no question of "multi-hop" paths and since there is a unique key shared between any two nodes, the computational complexity decreases as well.

Thus, in a DSN, it is important to study the subset of nodes (clique, in graph theoretic terminology) that are connected to each other. By connectivity of two nodes we mean that the nodes share one or more common secret key(s) for secure communication. In this paper we study the basic combinatorial designs [8] and their extensions in terms of merging [3, 4] to estimate the cliques of maximum size. We show that if one uses a $(v = rk, b = r^2, r, k)$ configuration, where each block corresponds to a node [8], then the maximum clique size is $r = \sqrt{b}$. We also study the extension of the basic design where a few blocks are merged to get a node [3,4] and show that in such a strategy the clique size becomes considerably larger than what is available in the basic design [8].

2. Preliminaries

2.1 Basics of Transversal Design

Let A be a finite set of subsets (also known as blocks) of a set X. A *set system* or *design* is a pair (X, A). The degree of a point $x \in X$ is the number of subsets containing the point x. If all subsets/blocks have the same degree k, then (X, A) is said to be uniform of rank k. If all points have the same degree r, (X, A) is said to be regular of degree r.

A regular and uniform set system is called a $(v,b,r,k) - 1$ design, where $|X| = v$, $|A| = b$, r is the degree and k is the rank. The condition $bk = vr$ is necessary and sufficient for existence of such a set system. A $(v,b,r,k) - 1$ design is called a (v,b,r,k) configuration if any two distinct blocks intersect in zero or one point.

A (v,b,r,k, λ) BIBD is a $(v,b,r,k) - 1$ design in which every pair of points occurs in exactly λ many blocks. A (v,b,r,k) configuration having deficiency $d = v - 1 - r(k - 1) = 0$ exists if and only if a $(v,b,r,k, 1)$ BIBD exists.

Let g, u, k be positive integers such that $2 \le k \le u$. A group-divisible design of type g^u and block size k is a triple $(X, \mathcal{H}, \mathcal{A})$, where X is a finite set of cardinality gu, $\mathcal{H}$ is a partition of X into u parts/groups of size g, and $\mathcal{A}$ is a set of subsets/blocks of X. The following conditions are satisfied in this case:

1. $|H \cap A| \le 1 \forall H \in \mathcal{H}, \forall A \in \mathcal{A}$,
2. every pair of elements of X from different groups occurs in exactly one block in $\mathcal{A}$.

A Transversal Design $TD(k, n)$ is a group-divisible design of type n^k and block size k. Hence $H \cap A = 1 \forall H \in \mathcal{H}, \forall A \in \mathcal{A}$.

Let us now describe the construction of a transversal design. Let p be a prime power and $2 \le k \le p$. Then there exists a $TD(k, p)$ of the form $(X, \mathcal{H}, \mathcal{A})$ where $X = \mathbb{Z}_k \times \mathbb{Z}_p$. For $0 \le x \le k - 1$, define $H_x = \{x\} \times \mathbb{Z}_p$ and $\mathcal{H} = \{H_x : 0 \le x \le k-1\}$

For every ordered pair (i, j) (1), define a block $A_{i,j} = \{x, (ix + j) \bmod p : 0 \le x \le k - 1\}$. $(i, j) \in \mathbb{Z}_p \times \mathbb{Z}_p$

In this case, $\mathcal{A} = \{A_{i,j} : (i, j) \in \mathbb{Z}_p \times \mathbb{Z}_p\}$. It can be shown that $(X, \mathcal{H}, \mathcal{A})$ is a $TD(k, p)$.

Now let us relate a $(v = kr, b = r^2, r, k)$ configuration with sensor nodes and keys. X is the set of $v = kr$ number of keys distributed among $b = r^2$ number of sensor nodes. The nodes are indexed by $(i, j) \in \mathbb{Z}_r \times \mathbb{Z}_r$ and the keys are indexed by $(i, j) \in \mathbb{Z}_k \times \mathbb{Z}_r$. Consider a particular block $A_{\alpha,\beta}$. It will contain k number of keys $\{(x, (x\alpha + \beta) \bmod r) : 0 \le x \le k - 1\}$. Here $|X| = kr = v$, $|\mathcal{H}_x| = r$, the number of blocks in which the key (x, y) appears for $y \in \mathbb{Z}_r$, $|A_{i,j}| = k$, the number of keys in a block. For more details on combinatorial design refer to [9, 8].

Note that if r is a prime power, we will not get an inverse of $x \in \mathbb{Z}_r$ when x is not a unit of $\mathbb{Z}_r$ i.e., $\gcd(x, r) > 1$. This is required for key exchange protocol. So basically we should consider the field $GF(r)$ instead of the ring $\mathbb{Z}_r$. However, there is no problem when r is a prime by itself. In this paper we generally use $\mathbb{Z}_r$ since in our examples we consider r to be prime.

2.2 Lee-Stinson Approach

Consider a $(v = rk, b = r^2, r, k)$ configuration. There are $b = r^2$ many sensor nodes, each containing k distinct keys. Each key is repeated in r many nodes. Also v gives the total number of distinct keys in the design. One should note that $bk = vr$ and $v - 1 > r(k - 1)$. The design provides 0 or 1 common key between two nodes. The design ($v = 1470$, $b = 2401$, $r = 49$, $k = 30$) has been used as an example in [8]. The important parameters of the design are as follows.

The expected number of common keys between any two nodes is $p_1 = \frac{k(r-1)}{b-1} = \frac{k}{r+1}$. In the given example, $p_1 = \frac{30}{49+1} = 0.6$.

There is a good proportion of pairs (40%) with no common key, and two such nodes will communicate through an intermediate node. Assuming a random geometric deployment, the example shows that the expected proportion such that two nodes are able to communicate either directly or through an intermediate node is as high as 0.99995.

Under adversarial situation, one or more sensor nodes may get compromised. In that case, all the keys present in those nodes cannot be used for secret communication any longer, i.e., given the number of compromised nodes, one needs to calculate the proportion of links that cannot be used further. The expression for this proportion is $fail(s) = 1 - \left(1 - \frac{r-2}{b-2}\right)^s$, where is the number of nodes compromised. In this particular example, $fail(10) = 0.17951$. That is, given a large network comprising as many as 2401 nodes, if 10 nodes are compromised, almost 18% of the links become unusable.

3. Analysis of Clique Sizes

First we study the maximum clique size where the $(v = rk, b = r^2, r, k)$ configuration is used and each block in the design corresponds to a sensor node, which is the idea proposed in [8].

Theorem 1. Consider a DSN with b many nodes constructed from a $(v = rk, b = r^2, r, k)$ configuration. The maximum clique in this case is of size r.

Proof. First we prove that there is a clique of size r. It is known that a key is repeated in r many different blocks. Fix a key. Thus, there are r many distinct blocks which are connected to each other by the fixed key. Hence there is a clique of size r.

Now we prove that there is no clique of size $r + 1$, because that will rule out the possibility of cliques of larger size. Let there be a clique of size $r + 1$. Note that the (v, b, r, k) configuration results from $TD(k, r)$ (see Subsection 2.1). In this case each block is identified by two indices (i, j), $0 \le i, j \le r - 1$. Further two blocks having same value of i (i.e., in the same row) can't have a common key. The moment one chooses $r + 1$ blocks, at least two of the blocks must be from the same row (by pigeon hole principle as there are at most r many rows) and are disjoint, which is a contradiction to the basic assumption of a clique having size $r + 1$.

It should be observed that the clique size r is exactly the square-root of the number of nodes $b = r^2$. Note that in such a case two nodes/blocks either share a common secret key or not. Consider the graph with b^2 many nodes/vertices where each block corresponds to a node. Now two vertices are connected by an edge if they share a common secret key, otherwise they are not connected. Now a block contains k many distinct keys. For each key, a clique of size r is formed. Thus a vertex/node in this graph participates in k many cliques each of size exactly r.

Given two keys, which never occur together in the same block, will form cliques which are completely disjoint. On the other hand, two keys may occur together at most in a single block. In such a case, the two different cliques generated by them can intersect on a single node/vertex corresponding to the block that contains both the keys.

3.1 The Merging Approach

To overcome certain restrictions in the strategy provided in [8] (explained in the previous subsection), we have provided a strategy to merge certain blocks to construct a sensor node [3, 4]. The basic idea is to start from a $(v = rk, b = r^2, r, k)$ configuration. Then we merge z many blocks to form a single sensor node. Thus the maximum number of sensor nodes available in such a strategy is $\left\lfloor \frac{r^2}{z} \right\rfloor$. We have studied a random merging strategy in [3], where randomly chosen z many blocks are merged to get a sensor node. In such a scenario, we found that the number of common keys among any two nodes approximately follows the binomial distribution $\mathcal{B}\left(z^2, \frac{k}{r+1}\right)$. The expected number of common secret keys among any two nodes is $\frac{z^2 k}{r+1}$ (see [3, Theorem 1] for more details). It has been shown that this strategy provides favorable results compared to [8]. Note that in [3], the blocks are merged randomly. So it may happen that the blocks being merged may have common secret key(s) among themselves. This is actually a loss, since we really do not need a common key among the blocks that are merged to get a single node. Hence, in [4], we improved the strategy such that only disjoint blocks are merged to construct node. This provides little better parameters compared to [3]. In this paper we will show that our strategy [3,4] provides better clique size than that of the design presented in [8].

Now we concentrate on the cliques where blocks are merged to get a node [3, 4]. It is worth mentioning that the number of blocks is $\left\lfloor \frac{r^2}{z} \right\rfloor (1)$ in this case. From [3, Theorem 1], each key will be present in Q many nodes, where average value of Q is

$\hat{Q} = \frac{1}{kr}\left(\left\lfloor \frac{b}{z} \right\rfloor\right)\left(zk - \binom{z}{2}\frac{k}{r+1}\right) \simeq r$. So cliques of size $\simeq r$ are available in the design where merging strategy is employed.

We like to highlight that the value of z is much less than r (as example, r = 101, z = 4) though it is not a serious restriction in the proof of our results in the following discussion.

Thus we like to point out the following improvement in the merging strategy over the basic technique.

1. In the basic design, there are r^2 many nodes (each block corresponds to a sensor node) and the maximum clique size is r.
2. Using the merging strategy, there are $\left\lfloor \frac{r^2}{z} \right\rfloor$ many nodes (z many blocks are merged to get a sensor node) and the maximum clique size is $\simeq r$. Thus there is an improvement by a factor of $\sqrt{z}$ in the size of clique.

Let us present some examples to illustrate the comparison. The design ($v = 1470$, $b = 2401$, $r = 49$, $k = 30$) has been used as an example in [8]. Hence there are 2401 nodes and the largest clique size is 49. Now consider a ($v = 101 \cdot 7$, $b = 101^2$, $r = 101$, $k = 7$) configuration and merging of $z = 4$ blocks to get a node. Thus there will be 2550 (we take this value as it is comparable to 2401) many nodes. We have cliques of size $\simeq 101$ on an average, which shows the improvement.

Next we provide a more improved result by increasing the clique size beyond r. We present a merging strategy where one can get a clique of size $r + z - 1 \geq r$ for $z \geq 1$. The result is as follows.

Theorem 2. Consider a (v,b,r,k) configuration with $b = r^2$. We merge z many blocks to form each node in achieving a DSN having $N = \left\lfloor \frac{b}{z} \right\rfloor$ many sensor nodes. Then there exists an initial merging strategy which will always provide a clique of size r + z − 1.

Proof. Let's denote the nodes by $v_1, v_2, \ldots$. Initially choose the first column of the $TD(k, r)$ and place the r blocks (indexed by $(i, 0)$ for $0 \leq i \leq r - 1$) successively to fill up the first slot (out of the z slots) of the first r nodes $v_1, v_2, \ldots, v_r$. That will obviously yield a clique of size r as any two blocks in a specific column always share a common key.

The rest of the available blocks will always be traversed in column-wise manner. That is the next available block is now the one indexed by (0, 1). Let us refer to the next available block by (i, j) for the rest of the present discussion. Once a block is used, we apply the update function on its index to get the next available node. Update (i, j) to $((i + 1) \bmod r, j + \delta)$, where $\delta = 0$, if $i < r - 1$ and $\delta = 1$ when $i = r - 1$.

We go on adding new nodes for $t = 1$ to $z - 1$ to generate a clique of size $r + z - 1$ at the end.

To add a new node v_{r+t}, proceed as follows. Choose the first available block (i, j) and put it in v_{r+t}. Place the next available blocks in $v_1, v_2, \ldots, v_k$ as long as $i \leq r - 1$. After using the last element of current column, the update function provides the first block of the next column. In that case, we add this new block $(0, j)$ to the node v_{r+t}. Then again the next available blocks are put into the nodes $v_{k+1}, v_{k+2}, \ldots$, in the similar manner. Once the blocks in that column gets exhausted, we again add the first block of the next column to v_{r+t} and the following blocks to the nodes as long as we

reach v_{r+t-1}. Thus it is clear that all the nodes $v_1,\ldots,v_{r+t-1}$ are connected to v_{r+t} increasing the size of the clique by 1.

In this strategy, the value of t is bounded above by $z-1$ as otherwise the number of blocks in a node will increase beyond z. The remaining blocks will be arranged randomly to have z blocks in each node to get $\left[\frac{r^2}{z}\right]$ many nodes in completing the merging strategy.

Now we present an example corresponding to the strategy presented in Theorem 2.

Example 1. Consider the $TD(k,r = 25)$. Let $z = 2$. Consider the 5^2 blocks of the TD arranged in the form of a 5×5 matrix. If we adopt the strategy outlined in the proof of Theorem 2, initially, the following clique is obtained: $v_1 \to \{(0,0)\}$, $v_2 \to \{(1,0)\}$, $v_3 \to \{(2,0)\}$, $v_4 \to \{(3,0)\}$, $v_5 \to \{(4,0)\}$. Next (0,1) is put in the new node v_6 and then (1,1) is added to v_1, (2,1) is added to v_2, (3,1) is added to v_3, (4,1) is added to v_4. As the second column gets exhausted, (0,2) is added to the new node v_6 and then (1,2) is added to v_5. Thus we get, $v_1 \to \{(0,0),(1,1)\}$, $v_2 \to \{(1,0),(2,1)\}$, $v_3 \to \{(2,0),(3,1)\}$, $v_4 \to \{(3,0),(4,1)\}$, $v_5 \to \{(4,0),(1,2)\}$, $v_6 \to \{(0,1),(0,2)\}$ and they form a clique of size 6.

Next we observe that the clique size we present in Theorem 2 is not the maximum achievable one. One can indeed find a different merging strategy that provides a clique of larger size. Here is an example.

Example 2. Taking a different arrangement compared to Example 1, we get a clique of size 7 as follows: $v_1 \to \{(0,0),(2,1)\}$, $v_2 \to \{(1,0),(3,1)\}$, $v_3 \to \{(2,0),(4,1)\}$, $v_4 \to \{(3,0),(0,2)\}$, $v_5 \to \{(4,0),(1,2)\}$, $v_6 \to \{(0,1),(2,2)\}$, $v_7 \to \{(1,1),(3,2)\}$.

Thus it will be interesting to devise a merging strategy which will provide the largest clique size when the (v,b,r,k) configuration and z are fixed.

Theorem 3. Consider a (v,b,r,k) configuration with $b = r^2$. We merge z many blocks to form each node in achieving a DSN having $N = \left[\frac{b}{z}\right]$ many sensor nodes. Then there exists an initial merging strategy which will always provide a clique of size $2r - \left(\left[\frac{r}{z}\right]\right)$

Proof. Let the nodes forming the clique be $v_1, v_2, \ldots, v_{r+t}$. Each node has z number of empty slots. These slots are to be filed up by the properly chosen blocks. The TD may be considered as a $r \times r$ matrix filled with the values 0 to $r^2 - 1$. The value at the (i,j)-th position of the matrix is $i \cdot r+j$ and that entry is used as an identification of the corresponding block in the TD. We try to form a clique with $r + t$ nodes, $t < r$. The value of t will be clearer as we go through the proof.

A column in the matrix (corresponding to the TD) is chosen first and the r blocks are placed one by one in r blank nodes viz., $v_1, v_2, \ldots, v_r$. As all the blocks in the same column share the same secret key, the nodes $v_1, v_2, \ldots, v_r$ form a clique.

Then another column is chosen and the blocks are placed in the next t nodes, one each. In other words, the blocks are put in the nodes $v_{r-1}, v_{r+2}, \ldots, v_{r+t}$ and they form a clique among themselves. The rest $r - t$ blocks are added in the first $r - t$ nodes, viz., $v_1, v_2, \ldots, v_{r-t}$ (in the second slot). Thus each of $v_1, v_2, \ldots, v_{r-t}$ gets connected to each of $v_{r+1}, v_{r+2}, \ldots, v_{r+t}$.

In a similar fashion, the third column is chosen, and the blocks are placed in $v_{r+1}, v_{r+2}, \ldots, v_{r+t}$, one each (in the second slot). The rest $r - t$ blocks are placed in $v_{r-t+1}, v_{r-t+2}, \ldots, v_{r-t+r-t}$ (in the second slot). Thus each of $v_{r-t+1}, v_{r-t+2}, \ldots, v_{r-t+r-t}$ gets connected to each of $v_{r+1}, v_{r+2}, \ldots, v_{r-t}$.

We will continue the above process as long as the second slots of the first r nodes are eventually filled up. However, the continuation can be performed at most z many times as a node may accommodate at most z blocks. Each time $r-t$ new nodes are connected out of a target of r nodes. Thus, in order to complete the above process, we must have $r \le z(r-t), t \le r - \lceil \frac{r}{z} \rceil$.

Example 3. The technique in Theorem 3 outputs cliques of size $r + t$ where $t = r\left(\lceil \frac{r}{z} \rceil\right)$. Let us consider $r = 5$ and $z = 2$ as in the previous examples. This technique outputs a clique of size $5 + t$ where $t = 5 - \left(\lceil \frac{5}{2} \rceil\right) = 2$, i.e., we get a clique of size 7.

The technique constructs the clique as follows:

$v_1 \to \{(0,0),(3,1)\}$, $v_2 \to \{(1,0),(4,1)\}$, $v_3 \to \{(2,0),(3,2)\}$, $v_4 \to \{(3,0),(4,2)\}$, $v_5 \to \{(0,1),(0,2)\}$, $v_6 \to \{(1,1),(1,2)\}$, $v_7 \to \{(2,1),(2,2)\}$,

Note that in the basic (v,b,r,k) configuration or after our merging strategy, the size of cliques are not dependent on the number of keys in each block/node. It is clear that the connectivity of the DSN increases with the increasing number of keys in each node. However, increasing the number of keys is constrained by the limited memory capacity of a sensor node. It is a nice property that the clique size does not increase with the number of keys in each node (using our strategy) as otherwise one may be tempted to obtain cliques of larger sizes by increasing the number of keys in each node (i.e., by increasing the edges in the graph).

3.2 Configurations Having Complete Block Graphs: Projective Planes

Since we are talking about cliques, we should also revisit the designs where the entire DSN forms a clique. In [8, Theorem 11, 12], it has been pointed out that the block graph of a set system is a complete graph if and only if the set system is the dual design of a BIBD and in particular, there exists a key pre-distribution scheme for a DSN having $q^2 + q + 1$ nodes, in which every node receives exactly $q + 1$ keys and in which any two nodes share exactly one key. It is also stated that such designs are not recommendable as a key pre-distribution scheme in large DSNs because of storage limitation in each sensor node. We like to point out that even if the storage space is not a limitation, then also this scheme is not suitable. The reason is as follows.

In this design any two nodes share a common key. However, for better resiliency one may like to have more common keys among any two nodes (this is one important motivation for our merging strategy [3,4]). Even if one maintains multiple keys against each identifier, the projective planes do not help because compromise of a single node results in discarding the identifiers contained in each node (block) and all the corresponding keys for each identifier also get discarded. Thus the resiliency measure *fail(s)*, (the probability that a given link is affected due to the compromise of s number of randomly chosen nodes) does not improve (i.e., does not reduce).

4. Conclusion

In this paper we consider the DSNs where the key pre-distribution mechanism evolves from combinatorial design. Such schemes provide the advantage of very low complexity key exchange facility (only inverse calculation in finite fields). In terms of distributed computing and communication among the sensor nodes, it is important to study the subset of nodes that are securely connected to each other (clique). In this paper we have studied

that in details. We studied the cliques corresponding to the (v,b,r,k) configuration where each block corresponds to a node. Further we study the scenario when more than one blocks are merged to generate a node. We show that the clique size gets improved in such a scenario. An interesting future work in this area is to implement a merging strategy such that one can get cliques of maximum size after the merging.

References

1. R. Blom, "An optimal class of symmetric key generation systems," *Eurocrypt* **84**, 335–338, LNCS 209, 1985.
2. S. A. Camtepe, and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor networks," *Eurosics,* Lecture Notes in Computer Science, Springer, **3193**, 293–308, 2004.
3. D. Chakrabarti, S. Maitra, and B. Roy, "A key pre-distribution scheme for wireless sensor networks: Merging blocks in combinatorial design," Accepted *in 8th Information Security Conference, ISC 2005,* (to be held in Singapore), Sept. 20–23, 2005.
4. D. Chakrabarti, S. Maitra, and B. Roy, *A hybrid design of key pre-distribution scheme for wireless sensor networks*. To be presented at the *The 1st International Conference on Information Systems Security,* ICISS (Kolkata, India). December 2005. Proceeding to be published in *Lecture Notes in Computer Science 3803,* Springer–Verlag.
5. W. Du, J. Ding, Y. S. Han, and P. K. Varshney " A pairwise key pre-distribution scheme for wireless sensor networks." in *Proceedings of the 10th ACM Conference on Computer and Communication Security,* ACM CCS, 42–51, 2003.
6. L. Eschenauer, and V. B. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ACM, CCS, 41–47, 2002.
7. J. Lee, and D. Stinson, *Deterministic key predistribution schemes for distributed sensor networks*. SAC, 2004.
8. J. Lee, and D. Stinson , "A combinatorial approach to key predistribution for distributed sensor networks," *IEEE Wireless Computing and Networking Conference (WCNC 2005)*, New Orleans, LA, USA, March, 13–17, 2005.
9. A. P. Street and D. J. Street, *Combinatorics of experimental design*. Oxford: Clarendon Press, 1987.

Dibyendu Chakrabarti received his Master of Technology in Computer Science in the year 1998 from the Indian Statistical Institute, Kolkata. Currently he is pursuing his Ph.D from the Indian Statistical Institute, Kolkata. He is working in the area of Sensor Networks.

Subhamoy Maitra received his Bachelor of Electronics and Telecommunication Engineering degree in the year 1992 from Jadavpur University, Kolkata and Master of Technology in Computer Science in the year 1996 from the Indian Statistical Institute, Kolkata. He has completed Ph.D from the Indian Statistical Institute in 2001. Currently he is an Associate Professor at the Indian Statistical Institute. His research interest is in Cryptology, Digital Watermarking, and Sensor Networks.

Prof.Bimal Roy obtained his Master's degree from the Indian Statistical Institute, Calcutta, India in 1979 and Ph.D. from University of Waterloo, Canada in 1982. He is currently a professor at the Indian Statistical Institute, Kolkata. His research area includes Cryptography, Security, Combinatorics etc. His special topics of interest are: Sensor Networks, Visual Cryptography, Hash Functions, and Stream Ciphers.