IACSIT International Journal of Engineering and Technology, Vol.2, No.2, April 2010
ISSN: 1793-8236

Hybrid Design of Scalable Key Distribution for Wireless Sensor Networks

T. Kavitha, Dr.D.Sridharan

Abstract—**A Sensor Network is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon of very close to it. The wireless communication employed by sensor network facilitates eavesdropping and packet injection by an adversary. Therefore security must be provided for sensor network to ensure secrecy of sensitive data. To achieve security, keys must be agreed upon by communication nodes.**

The main task is to safely distribute the shared keys to the sensor nodes. The solution to key distribution is such that, a pool of symmetric keys is chosen and a subset of the pool (key chain) is distributed to each sensor node. Two nodes that want to communicate search their key chain to determine whether they share a common key; if they don't share key in common then there may be a path, called key path, among these two nodes where each pair of neighboring nodes on this path have a key in common. This paper deals with hybrid design of key distribution which combines combinatorial approach and probabilistic approaches to select a key pool and key chain from the pool

Index Terms— **Security in Wireless Sensor Network, Hybrid key distribution, combinatorial design theory, key management.**

I. Introduction

A wireless sensor network (WSN) is composed of large number of sensor nodes with limited power, computation, storage and communication capabilities. Environments, where sensor nodes are deployed, can be controlled or uncontrolled. If the environment is known and under control, deployment may be achieved manually to establish an infrastructure. However, manual deployments become infeasible or even impossible as the number of the nodes increases. If the environment is uncontrolled or the WSN is very large, deployment has to be performed by randomly scattering the sensor nodes to target area. It may be possible to provide denser sensor deployment at certain spots, but exact positions of the sensor nodes can not be controlled. Thus, network topology can not be known precisely prior to deployment.

Since the network topology is unknown prior to deployment, a key pre-distribution scheme is required where keys are stored into ROMs of sensors before the deployment. The keys stored must be carefully selected so to increase the probability that two neighboring sensor nodes, which are within each other's wireless communication range, have at least one key in common. Nodes which do not share a key may communicate through a path on which each pair of neighboring nodes share a key. The length of this path is called *key-path* length. Average *key-path* length is an important performance metric and design consideration.

Common approach is to assign each sensor node multiple keys, *randomly* drawn from a *key-pool*, to construct a *key-chain* to ensure that either two neighboring nodes have a key in common in their key-chains, or there is a key-path. Thus, challenge is to decide on size of the key-chain and key-pool so that every pair of nodes can establish a session key directly or through a path. Key-chain size is limited by storage capacity of sensor nodes. Moreover, very small key-pool increases probability of key share between any pair of sensor nodes by decreasing security in that number of keys to be discovered by an adversary decreases. Similarly, very large key-pool decreases probability of key share by increasing the security.

This article is structured as follows. In the next section some related works are given. Section III deals about deterministic approach. Section IV emphasizes design of hybrid key distribution. Lastly, section V and VI covers analysis and conclusion respectively.

Manuscript received July 20, 2009

T. Kavitha, Research Scholar, Department of Electronics and communication, College of Engineering Guindy, Anna University. (email:haikavi18@yahoo.co.in).

Dr.D.Sridharan, Assistant Professor, Department of Electronics and communication, College of Engineering Guindy, Anna University. (email:sridhar@annauniv.edu)

II. Related Works

A. Random Key Pre-Distribution Scheme

In key setup phase [1], a large key-pool of KP keys and their identities are generated. For each sensor, k keys are randomly drawn from the key-pool KP without replacement. These k keys and their identities form the key-chain for a sensor node. In shared-key discovery phase, two neighbor nodes exchange and compare list of identities of keys in their key-chains. Basically, each sensor node broadcasts one message, and receives one message from each node within its radio range where messages carry key ID list of size k.

Cluster key grouping [2], scheme proposes to divide key chains into C clusters where each cluster has a start key ID. Remaining key IDs within the cluster are implicitly known from the start key ID. Thus, only key IDs for clusters are broadcasted during shared-key discovery phase which means messages carry key ID list of size c instead of k.

After shared-key discovery phase, some node pairs may not be able to find a key in common. These pairs apply path-key establishment phase to communicate securely through other nodes. Scalability and resilience of the solutions can be improved by using larger key pools. But, larger key-pool means smaller probability of key share

because key-chain size may not increase due to storage limitations. Probability that a link is compromised, when a sensor node is captured, is k/KP which is very high for small key-pools, and produces low resilience.

B. Q-Composite Random Key Pre-Distribution Scheme

Q-composite random key pre-distribution scheme [3] requires q common keys to establish a link key. Link key $K_{A,B}$ between a pair of sensor nodes SA and SB is set as hash of all common keys $K_{A,B}$ = Hash(K1||K2||K3|| . . . ||Kq). The scheme improves resilience because probability that a link is compromised, when a sensor node is captured, decreases from k/KP. But, probability of key sharing also decreases because a pair of nodes has to share q keys instead of one.

C. Polynomial Based Key Pre-Distribution Scheme

Polynomial based key pre-distribution scheme [4] distributes a polynomial share (a partially evaluated polynomial) to each sensor node by using which every pair of nodes can generate a link key. Symmetric polynomial P(x, y) (P(x, y) = P(y, x)) of degree k is used. The coefficients of the polynomial come from GF (q) for sufficiently large prime q. Each sensor node stores a polynomial with k + 1 coefficients which come from GF (q). Every pair of sensor nodes can establish a key. The solution is k-secure, meaning that coalition of less than k+1 sensor nodes knows nothing about pair-wise keys of others.

Polynomial pool-based key pre-distribution scheme[5] considers the fact that not all pairs of sensor nodes have to establish a key. It combines Polynomial based key pre-distribution scheme with the key pool idea to improve resilience and scalability. For key setup phase, a set F of k degree polynomials over finite field GF (q) is generated. Each sensor node Si receives a subset Fi of the polynomial set F (Fi (subset of) F).

D. Key Matrix Based Dynamic Key Generation

All possible link keys in a network of size N can be represented as an N × N key matrix. It is possible to store small amount of information to each sensor node, so that every pair of nodes can calculate corresponding field of the matrix, and uses it as the link key. This scheme[B] uses a public (Y + 1) × N matrix G and a private N × (Y+ 1) matrix D which is generated over GF(q) and where N is size of the network. Solution is Y-secure, meaning that keys are secure if no more than Y nodes are compromised. Matrix G must have (Y + 1) linearly independent columns to provide Y-secure property. Key matrix is a symmetric matrix K = $(D.G)^T$ .G. Sensor node Si stores $column_i$ of size Y+1 from matrix G as public information, and row_i of size Y+1 from matrix (D.G)T as private information. A pair of sensor nodes (Si, Sj), first exchange their public information $column_i$ and $column_j$. The link key is then generated as $K_{ij} = row_i \times column_j$ and $K_{ji} = row_j \times column_i$ respectively. The scheme requires costly multiplication of two vectors of size Y + 1 where the elements are as large as the corresponding cryptographic key size.

E. Multiple Space Key Pre-Distribution Scheme

Multiple space key pre-distribution scheme[6] improves the resilience of previous scheme. It uses a public matrix G and a set of W private matrices D. These matrices form W spaces (D_i, G) for i = 1. . . W. For each sensor node, a set of T spaces are randomly selected among these W spaces. Required keying materials for each selected space are stored to the sensor node as in previous scheme; therefore, each sensor node stores T +1 vectors of size Y+1. In shared key discovery phase, a pair of nodes first agrees on a common space for which nodes has to exchange an extra message which includes T space IDs. It is possible that a pair of nodes does not share a common space, in that case they have to apply path-key establishment phase to establish a key through intermediate nodes.

Scalability of key matrix based dynamic key generation scheme is improved in multiple space Blom's scheme (MBS). The scheme divides nodes into two sets U and V to form bipartite key connectivity graph. That means, not every pair of nodes has to share a key. Another difference from key matrix based dynamic key generation scheme is that private matrix D is not necessarily symmetric. Secret information $column^T{}_u D$ is assigned to each node. Nodes can exchange their public information to calculate secret key. Larger networks are supported by Deterministic multiple space Blom's scheme (DMBS). DMBS increases scalability with the cost of decreased resilience because capture of one sensor node compromises credentials of T- 1 other nodes.

III. Deterministic Approach

In this approach, the keys in the key chain can be determined. (i.e.) they are not selected randomly. Combinatorial design is a deterministic approach for key distribution.

Combinatorial design based pair-wise key pre distribution scheme is based on block design techniques in combinatorial design theory. It employs symmetric and generalized quadrangles design techniques.

A. Symmetric Design

Balanced Incomplete Block Design (BIBD is an arrangement of v distinct objects into b blocks such that each block contains exactly k distinct objects, each object occurs in exactly r different blocks, and every pair of distinct objects occurs together in exactly y blocks. The design can be expressed as (v, k, λ), or equivalently (v, b, r, k, λ), where: λ(v-1)=r(k-1) and bk=vr. A BIBD is called Symmetric BIBD or Symmetric Design when b=v [10].

A Finite Projective Plane consists of a finite set P of points and a set of subsets of P, called lines. For an integer q where q>=2, Finite Projective Plane of order q has four properties: 1) every line contains exactly q+1 points; 2) every point occurs on exactly q+1 lines; 3) there are exactly q2+q+1 points; and 4) there are exactly q2+q+1 lines. If we consider lines as blocks and points as objects, then a Finite Projective Plane of order q is a Symmetric Design with parameters (q2+q+1, q+1,1).

For a network of size N, q is selected such that q2+q+1 is greater than N. Then the parameters in finite projective plane are mapped to key distribution. Symmetric Design has a very nice property that any pair of blocks shares exactly one object. The probability of key share between any pair of nodes is Psym=1, so that Average Key-Path Length is 1. Resilience is an important security metric, but it contradicts with probability of key share because as more keys are shared between blocks more blocks are affected by compromise of a block. Thus, our symmetric algorithm provides better

probability of key share than probabilistic algorithms by sacrificing the resilience.

We first look at the amount of blocks to be captured to compromise the object set. There are two ways to capture nodes: selectively or randomly. In the case of selective capture, we may simply assume that the attacker has ability to monitor whole network and selects the nodes wisely. Since key-chain size is q+1 for a Symmetric Design with q2+q+1 nodes and keys, an attacker needs at least q+1 key-chain to be able to recover the key-pool. A wise attacker may select to capture the nodes which have the same specific key in their key-chains. From the properties of Symmetric Design, we know that there are such key-chains. Since every pair of keys can occur in exactly one key-chain then every q2+q keys must be pairing with the specific key in these q+1 key-chains. But, an unlucky attacker who selects the nodes randomly might be capturing q2 key-chains which do not include the specific key. Therefore, an unlucky attacker may need to capture q2 +1 key-chain to be able to recover the key-pool.

B. Generalized Quadrangle

A Finite Generalized Quadrangle GQ(s,t) is an incidence structure S=(P, B, I) where P and B are disjoint and nonempty sets of points and lines, respectively, and is a symmetric point-line incidence relation satisfying the following axioms:
1) Each point is incident with t+1 lines (t>=1) and two distinct points are incident with at most one line.
2) Each line is incident with s+1 points (s>=1) and two distinct lines are incident with at most one point.
3) If x is a point and L is a line not incident (I) with x, then there is a unique pair (y, M) € PXB for which x | M | y | L
Here also first q is selected and based on that and based on the generalized quadrangle chosen, the generalized quadrangle is constructed and then is mapped to the key distribution.

In a GQ(s,t), there are b=(t+1)(st+1)lines, and a line intersects with t(s+1) other lines. Thus, in a design generated from a GQ, a block shares an object with t(s+1) other blocks. An unlucky attacker may need to capture st(t+1) nodes before reaching a node which includes a specific key. Therefore, an unlucky attacker needs to capture st2+st+1 nodes to recover the key-pool. GQ(s,t) provides better resilience than symmetric Design.

IV. Design

Hybrid key distribution is the scheme that combines both probabilistic and deterministic approaches in order to utilize the advantages of both these methods. Here we use Symmetric design and combine it with the random approach. The design of this approach is carried out by first designing the Symmetric design and then Complementary of this design is found. Later these two are combined using random approach to provide better flexibility and scalability. There exist several methods for Symmetric design construction. Here we are using the Mutually Orthogonal Latin Squares (MOLS) to construct projective plane which is a subset of the Symmetric design.

A. Symmetric BIBD Design

Symmetric BIBD can be represented using three parameters (v, k, λ), where v is the number of distinct objects, k is the number of distinct objects in a single block and λ is the number of blocks in which every pair of distinct objects occurs. Here we are going to design only a finite projective plane which is a subset of Symmetric BIBD where the parameters are (q^2+q+1, q+1, 1) where q is a prime power[10]. Following are the steps involved in designing a finite projective plane.

1) Selection of q

The prime power q should be selected in such a way that q^2+q+1 is greater than N, where N is the number of sensor nodes. For example the value of q can be taken as 3 for a network size of 10 nodes. (3*3 + 3 + 1 > 10).

2) Construction of Mutually Orthogonal Latin Squares (MOLS)

A Latin square on q symbols is a q x q array such that each of the q symbols occurs exactly once in each row and in each column. The number q is called the order of square.
For example a Latin square of order 3 is

1	2	3
3	1	2
2	3	1

Let A = (a_{ij}) and B = (b_{ij}) are any two q x q arrays, the join of A and B is a q x q array whose (i,j)th element is the pair (a_{ij},b_{ij}). For example let L1 and L2 be two Latin squares of order 3.

L1 =				L2 =		
	1	2	3		2	3	1
	2	3	1		1	2	3
	3	1	2		3	1	2

Then the join operation would result in

(1, 2)	(2, 3)	(3, 1)
(2, 1)	(3, 2)	(1, 3)
(3, 3)	(1, 1)	(2, 2)

Latin squares A and B of order q are orthogonal if all entries of A join B are distinct. Latin squares L1, L2…… are MOLS if they are orthogonal in pairs. For example the above two Latin squares L1 and L2 are MOLS. Totally there will be q-1 MOLS for order q.

In our implementation we are going to construct MOLS directly. The implementation procedure is given below.
First label the rows and then the columns of a q x q square with the numbers 0, 1… q-1. It is convenient to assume that the numbers are listed in the same order for both rows and columns.

Next we use the linear polynomial f(x, y) =ax+y, where x and y represents the rows and columns respectively and 'a' is any value between 1 and q-1. The q-1 matrices are constructed by taking the value of 'a' to be 1, 2…q-1. These q-1 matrices represent the q-1 MOLS.

The arithmetic used here is Finite Field arithmetic. The field arithmetic for prime numbers (say q) is nothing more than addition and multiplication modulo the prime q.

3) Construction of Affine Plane

Affine plane can be represented using the parameters (q^2, q, 1). It can be constructed from the MOLS generated. Let the q-1 MOLS be represented as L_1… L_{Q-1}.The blocks of affine plane are constructed as follows.
Here x and k are variables…
For 1 <= x <= q-1, 1 <= k <=q, define
$A_{x,k} = \{ (i , j): L_x (i , j) = k \}$
For 1 <= k <=q, define
$A_{q,k} = \{ (k , j) : 1 <= j <= q \}$
And for 1 <=k <=q, define
$A_{q+1, k} = \{ (i , k) : 1 <= i <= q \}$
Finally, let

$A = \{A_{x,k}: 1 <= x <= q+1, 1 <= k <= q\}$, A is the Affine plane.

E.g. Orthogonal LS of order 3.

L_1= 1 3 2 L_2= 1 3 2
2 1 3 3 2 1
3 2 1 2 1 3

Blocks of Affine Plane

A_{11} = { (1,1) , (2,2) , (3,3) } A_{21} = { (1,1) , (2,3) , (3,2) }
A_{12} = { (1,3) , (2,1) , (3,2) } A_{22} = { (1,3) , (2,2) , (3,1) }
A_{13} = { (1,2) , (2,3) , (3,1) } A_{23} = { (1,2) , (2,1) , (3,3) }
A_{31} = { (1,1) , (1,2) , (1,3) } A_{41} = { (1,1) , (2,1) , (3,1) }
A_{32} = { (2,1) , (2,2) , (2,3) } A_{42} = { (1,2) , (2,2) , (3,2) }
A_{33} = { (3,1) , (3,2) , (3,3) } A_{43} = { (1,3) , (2,3) , (3,3) }

4) Construction of Projective Plane from Affine Plane

Finally the finite projective plane can be constructed from the Affine Plane that has been generated. The Affine plane can be converted into a projective plane by including a new block. Let A be the affine plane of order q, then introduce a new block B such that $B = \{\infty_1, \infty_2 \ldots \infty_{q+1}\}$. Where $\infty_1, \infty_2 \ldots \infty_{q+1}$ are elements that are not available in the affine plane blocks. And then the projective plane is constructed by including one element in each of the affine plane blocks i.e. $A` = A \cup \{\infty_i\}$.

E.G Construction of (13, 4, 1) BIBD from Affine Plane of order 3.

The blocks of affine plane are

A_{11}={(1,1),(2,2),(3,3)} A_{12}={(1,3),(2,1),(3,2)}
A_{13}={(1,2),(2,3),(3,1)} A_{21}={(1,1),(2,3),(3,2)}
A_{22}={(1,3),(2,2),(3,1)} A_{23}={(1,2),(2,1),(3,3)}
A_{31}={(1,1),(1,2),(1,3)} A_{32}={(2,1),(2,2),(2,3)}
A_{33}={(3,1),(3,2),(3,3)} A_{41}={(1,1),(2,1),(3,1)}
A_{42}={(1,2),(2,2),(3,2)} A_{43}={(1,3),(2,3),(3,3)}

Let B be the new block

B={ (4,1) , (4,2) , (4,3) , (4,4) }

Then the blocks of projective plane are given as

$A`_{11}$={(1,1),(2,2),(3,3)}U(4,1)
$A`_{12}$={(1,3),(2,1),(3,2)}U(4,1)
$A`_{13}$={(1,2),(2,3),(3,1)}U(4,1)
$A`_{21}$={(1,1),(2,3),(3,2)}U(4,2)
$A`_{22}$={(1,3),(2,2),(3,1)}U(4,2)
$A`_{23}$={(1,2),(2,1),(3,3)}U(4,2)
$A`_{31}$={(1,1),(1,2),(1,3)}U(4,3)
$A`_{32}$={(2,1),(2,2),(2,3)}U(4,3)
$A`_{33}$={(3,1),(3,2),(3,3)}U(4,3)
$A`_{41}$={(1,1),(2,1),(3,1)}U(4,4)
$A`_{42}$={(1,2),(2,2),(3,2)}U(4,4)
$A`_{43}$={(1,3),(2,3),(3,3)}U(4,4)
B`={ (4,1) , (4,2) , (4,3) , (4,4) }

Totally we get 13 blocks. These blocks represent the symmetric design blocks. We can verify by checking whether it satisfies the properties of symmetric design. Here q=3, so we get totally 13 blocks (q^2+q+1) and each of the block contains 4 (q+1) elements. And there exists only one element in common between any two blocks thus satisfying the property (q^2+q+1, q+1, 1). Hence it is a symmetric design.

B. Complementary Design

Given a block design with a set of D = (v, k, λ) with a object set (key pool) S of |S|=v objects and blocks (key chains) $B_1, B_2 \ldots B_b$ where each block contains exactly k objects, Complementary Design has the complement blocks $ß_i = S\text{-}B_i$ as its blocks for $1 <= i <= b$, where i is any variable.

Đ is a block design with parameters (v, v-k, v-2k+λ). If D is a Symmetric Design, then Đ is also a Symmetric Design.

For example,

Let S = { 1 , 2 , 3 , 4 , 5 , 6 , 7 } , v = 7
B = {1,2,3} , {1,4,5} , {1,6,7} , {2,4,6} , {2,5,7} , {3,4,7} , {3,5,6}

Then the complement blocks are,

ß = {4,5,6,7} , {2,3,6,7} , {2,3,4,5} , {1,3,5,7} , {1,3,4,6} , {1,2,5,6} , {1,2,4,7}

C. Hybrid Design

In Hybrid Design we just combine the symmetric design and its complement using random approach. Let N be the number of sensor nodes. First we need to select the value of q (prime power) such that $2*(q^2+q+1) >= N$ (Here we note that in symmetric Design we selected q based on the condition q^2+q+1, and so we get a higher value of q in symmetric design). Next for the selected q value construct q-1 Mutually Orthogonal Latin Squares (MOLS) and then convert them into affine plane blocks. Then projective plane is constructed by embedding the affine planes.

Next step is to find the complementary for all the blocks in the projective plane. Let the number of blocks in projective plane be R (R<N). These R blocks are assigned to R sensor nodes and for the remaining N-R nodes we need to select the blocks from complementary design randomly. Here the selection of q plays a critical part. If the selected value is very less (i.e. nearer to N) then we get a hybrid design with less key chain size and less probability of key share. If the value of q is very high it almost has the same characteristics of the symmetric design.

D. Random Design

In Random key distribution, the keys for each of the key chain are randomly distributed. We can design a Random key distribution scheme with any key pool size and key chain size, so first we need to decide upon the following parameter values.

1) Length of the key pool

For our comparison purposes we can select the length to be equal to the key pool size of hybrid design.

2) Length of the key chain

For our comparison purposes we can select the length to be equal to the key chain size of hybrid design.

After selecting the values of these parameters we need to randomly select the keys for each of the key chains. Let n be the length of key pool, then the key identities are 1, 2…, n. Let the length of the key chain be t and number of key chains be c, and then we need to group the key identities into c groups of length t randomly.

E. Mapping

Mapping from Symmetric, Hybrid designs to key distribution is given in the Table I

TABLE I. MAPPING FROM COMBINATORIAL DESIGN TO KEY DISTRIBUTION

Symmetric and Hybrid Designs		Key Distribution
Object Set	→	Key-Pool (P)
Object Set Size v	→	Key-Pool Size (|P|)
A Block	→	A Key-Chain

# Blocks b	→	# Key-Chains
# Objects in a Block	→	# Keys in a Key-Chain (K)
# Blocks that an Object is in	→	# Key-Chains that a Key is in
Two Blocks share λ Object	→	Two Key-Chains share (x) Keys

V. Analysis

This project is implemented using the programming language java. Implementation is carried out in a system with an optimum hard disk space since the size of the database, to hold the key chains, is very high. In the simulation generated, the path that we have discovered is the shortest possible distance between the source and destination.

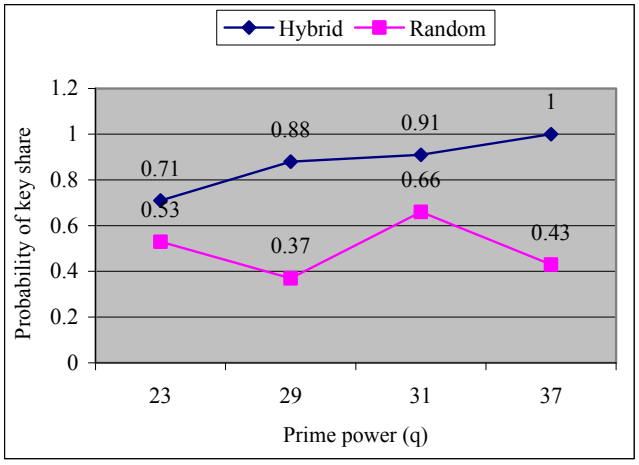

Fig 1 Probability of key share comparison graph

The performance of the three approaches, Random, Symmetric and Hybrid, are analyzed and compared. First we need to construct Random approach which is nothing but creating keys randomly for every key chain. Then the probability of key share (the probability by which every two nodes share a common key) is calculated for each of the three approaches. The probability of key share is always 1 for symmetric approach since every two nodes share a common key. In random approach the probability of key share is purely random and in hybrid approach it depends on the value of q being selected which is shown in Fig 1. If the q value is large then we get a high probability of key share.

Then the next comparison is based on the network size it can support. The network size supported by symmetric design is given by q^2+q+1, where q is the selected prime power. The network size supported by hybrid design is given by $2*(q^2+q+1)$. Hence hybrid design is more scalable and flexible than symmetric design. Hybrid design provides shorter average key path length. It also improves resilience of underlying symmetric design

VI. Conclusion

In this work, we have presented novel approaches to the key distribution problem in large scale sensor networks. In contrast with prior work, our approach is combinatorial based on Combinatorial Block Designs. We showed how to map from two classes of combinatorial designs to deterministic key distribution mechanisms. We remarked the scalability issues in the deterministic constructions and proposed hybrid mechanisms. Hybrid constructions combine a deterministic core design with probabilistic extensions to achieve key distributions to any network sizes. The analysis and computational comparison to the randomized methods show that the Hybrid approach has clear advantages: 1) it increases the probability of a pair of sensor nodes to share a key, and 2) decreases the average key-path length while providing scalability.

In this project we used hybrid mechanism, to determine the key chains, which has a probability of key share less than 1. So we can try to increase the probability to 1 by using various other mechanisms. Although the running time of our construction algorithm is far less when compared to that of other combinatorial designs it can further be reduced by adopting different techniques for constructing mutually orthogonal Latin squares.

References

[1] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *Proc. 9th ACM Conf. Comp. and Commun. Security*, Nov. 2002, pp. 41–47.

[2] Hwang, D., Lai, B., and Verbauwhede, I. "energy-memory-security tradeoffs in distributed sensor networks", *Proc 3rd International Conference on Ad-Hoc Networks and Wireless (ADHOC NOW 2004)* 2004.

[3] H. Chan, A. Perrig, and D. Song, "Random Key Pre-distribution Schemes for Sensor Networks," *IEEE Symp.Security and Privacy*, May 2003. pp. 197–213,

[4] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences", *Proc. 12th Annual Int'l. Cryptology conf* Advances in Cryptology - Crypto'92, LNCS 740, 1993, pp. 471–486.

[5] D. Liu and P. Ning. "Establishing Pairwise Keys in Distributed Sensor Networks," *Proc. 10th ACM Conf. Comp. and Commun. Securit*, Oct. 2003, pp. 52–61

[6] W. Du, J. Deng, Y. S. Han, P. K. Varshney, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks", *10th ACM Conference on Computer and Communications Security (CCS)*, Washington DC, Oct 2003, pp. 42–51.

[7] J. Lee and D. R. Stinson, "A Combinatorial Approach to Key Predistribution for Distributed Sensor Networks," *Proc. IEEE Wireless Commun. and Net. Conf.*, 2005.

[8] J. Lee and D. Stinson, "Deterministic Key Predistribution Schemes for Distributed Sensor Networks," *Selected Areas in Cryptography*, 2004

[9] Seyit A. Çamtepe, and Bülent Yener, "Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks", IEEE/ACM Transactions On Networking, Vol. 15, No. 2, April 2007, pp.345 – 358.

[10] I. Anderson, *Combinatorial Designs: Construction Methods*. Chicester, U.K.: Ellis Horwood, 1990.

T.Kavitha received Bachelor of Engineering in Electronics and Communication Engineering from Bharathidasan University in the year 2000 and Master of Engineering in Systems Engineering and Operations Research from College of Engineering Guindy, Anna University Chennai in the year 2006. Presently she is a research scholar in Anna University Chennai, India. Her area of interest includes Network security and wireless sensor networks.

Dr.D.Sridharan received B.Tech and M.E in Electronics Engineering in the year 1991 & 1993 respectively from Madras Institute of Technology, Anna

University and Ph.D degree in the department of Faculty of Information and Communication Engineering from Anna University in the year 2005. He was awarded by the Young Scientist Research Fellowship by SERC of Department of Science and Technology, Government of India.

He is currently working as an Assistant Professor, Department of Electronics and Communication Engineering, at College of Engineering Guindy, Anna University – Chennai, India. His research interest includes Internet Technology, Network Security, Distributed Computing, and VLSI for Wireless Communication.

He has published more than 25 papers in National/ International conferences and journals. He has visited USA, Italy, Germany, Singapore, Hong Hong and Dubai to participate and present research papers and he has also attended number of workshops sponsored by UNEFCO. He is a life member of Institute of Electronics and Telecommunication Engineers (IETE), Indian Society for Technical Education (ISTE), and Computer Society of India (CSI). Presently he is working on research project on Wireless Sensor Network sponsored by Department of Atomic Energy.