iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** VII **Month of publication:** July 2018

DOI: http://doi.org/10.22214/ijraset.2018.7142

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VII, July 2018- Available at www.ijraset.com

A Survey on Wireless Sensor Network Key Pre-Distribution

Dr. V. Umadevi [1], P. Shobiya [2]
[1]*Research Advisor, Jairam's Arts and Science College,Karur, Tamilnadu,India*
[2]*Research Scholar, Jairam's Arts and Science College,Karur, Tamilnadu,India*

Abstract: ***We discuss our proposed key pre-distribution mechanism using BCH code. We have mapped the BCH code to key identifier and the key corresponding to each key identifier are installed into the sensor nodes before deployment. We have found that our proposed scheme has a better resiliency and required the same or less number of keys to be stored in each sensor for a given number of nodes than the existing well known schemes. Our proposed scheme is also scalable too, in the sense, the addition of new nodes into the network does not require alteration, addition or modification of keys in the nodes present in the network. In this proposition we proposed a deterministic key pre-dissemination conspire utilizing BCH codes. We mapped the BCH code to key identifier and the keys comparing to each key identifier are introduced into the sensor hubs before organization. We contrasted our proposed conspire and existing one and found that it has a superior flexibility. Our proposed conspire is adaptable and requires the same or less number of keys for a given number of hubs than the current understood plans. We have additionally proposed an efficient key repudiation method utilizing a novel conveyed voting component in which neighbouring hubs of a sensor can vote against it in the event that they speculate the hub to be a traded off one.***
Keywords: Pairwise key, BCH, Code Polynomial, RR Scheme, CY Scheme

I. INTRODUCTION

We have already told that key pre-distribution in sensor network is a challenging task. Eschenaur and Gligor [1] was the first to address a probabilistic solution to this problem. Then Camtepe and Yener [2,3], Lee and Stinson [4,5] and many others proposed deterministic solution to this problem with the help of design theory. Ruj and Roy [6] was the first to provide a solution using Reed-Solomon code. They first use the coding theory as a deterministic solution to key pre-distribution. Chan Perrig and Song [7] modified Eschenauer and Gligor scheme. According to their q-composite scheme two nodes must share at-least q number of keys to have a secure path between them. The path key will be formed by the hash of all the common keys. Though for small number of node capture, resiliency was improved, the resiliency was a ected drastically as number of captured nodes increases.

The pairwise key scheme of Liu and Ning [8] is based on the polynomial pool based key pre-distribution by Blundo et. al. [9]. They have shown the calculation for the probability that two nodes share a common key. They have also shown the probability that a key is compromised. Later it was extended in [10] where they modi ed the scheme into a hypercube based key pre-distribution.

Zhu, Xu, Setia and Jajodia [11] also proposed a random pairwise scheme based on probabilistic key sharing where two nodes can establish shared keys without the help of an online KDC and only knowing each other's key id. Communication overhead in this scheme is very low. But if any node in the path is compromised then the key establishment process has to be restarted.

Liu, Ning and Du observed that sensor nodes in the same group are usually close to each other and they proposed a group based key pre-distribution scheme without using deployment knowledge [12, 13]. To overcome the problems of Liu et al's scheme [14], Martin Paterson and Stinson [13] proposed a group based design using resolvable transversal designs. To increase the cross group connectivity, they proposed that each node is contained in m cross groups rather than one. Though some additional storage is required. They did not give any algorithm for the construction of such designs.

II. KEY PRE-DISTRIBUTION USING BCH CODE

A. A Code is a Pair (q,c) Such That the Following Properties Are Satisfied

1) Q is a set of symbols.

2) C is a set of d-tuples of symbols called codeword where d 1 and d is an integer.

3) A code is said to be a linear code if it posses the following properties:

Sum of any two codewords belonging to same code is also a valid codeword belonging to that code, All zero codeword is always a valid codeword, and Minimum hamming distance will be the minimum weight of any non zero codeword.

A code is said to be a cyclic code if it is linear and any cyclic shift of a code-word is also a codeword belonging to the same code.

©IJRASET: All Rights are Reserved

BCH code is a cyclic linear block code, constructed from an alphabet set P. Length of the codewords are n = p^m - 1 where m is an integer and jPj= p. A generator polynomial g(x) is used to derive the codewords. Total number of possible codewords for an alphabet set P is p^k where k = n deg(g(x)). Here deg (g(x)) represents the highest degree of x in the generator polynomial g(x). The process of nding the generator polynomial for a particular code is described in Section 4.

Galois field, GF, is a field with nite number of elements. GF(q) has q number of elements. A GF of order q^m, that is GF(q^m), can be constructed from GF(q) where m is an integer. In such cases, GF(q) is called base field and GF(q^m) is called extension field.

Primitive polynomial f(x) over any Galois field is a prime polynomial over that GF with the property that in the extension field constructed from modulo f(x), every element of the extension field except zero can be expressed as a power of x.

If GF(q) is the base field and GF(q^m) is the extension field, then x^n 1 can be factorized over GF(q) where n = q^m 1. Let say, x^n 1 = $f_1(x)f_2(x)$ .....$f_p(x)$.

In the extension field, x^n 1 = (x $_j$) where $_j$ are all the non-zero elements of GF(q^m). Here, we can say that each $_j$ is a solution of exactly one of the $f_i(x)$. This $f_i(x)$ is called the minimal polynomial of the corresponding $_j$.

Set of elements in the extension field sharing the same minimal polynomial over base field are called conjugates with respect to GF(q).

If f(x) is the minimal polynomial of an element,say , in the extension field then the conjugate set including will be (^{q}; ^{q2} ; ::::::::::::; $^{qr\ 1}$) where $^{qr\ 1}$ = for any integer r.

III. PROPOSED WORK

The proposed scheme is scalable. Key pre-distribution in the pro-posed scheme is carried out in two phases. Fifirst phase consists of the construction of BCH codewords. In the second phase, we derive the key identifiers for each sensor from the BCH codeword. Each node is represented by means of a unique node polynomial derived from GF(p). The codeword is obtained in the Fifirst phase. Then in the second phase identifiers for each node is derived from the codeword obtained in the Fifirst phase. We describe below the two phases in key pre-distribution.

A. First Phase

The following steps are carried out in this phase to construct BCH codewords.

Step 1 : Choose the length of the codeword, n, such that n = p^m 1 where p is a prime or a prime power, and m is an integer.

Step 2 : Choose a primitive polynomial over GF(p) of degree m and construct GF(p^m).

Step 3 : Find the set of conjugates from the elements of the GF(p^m). For each set of conjugates nd the minimal polynomial corresponding to that set.

Step 4 : Choose a value, t, which is the maximum number of errors BCH code can correct. The generator polynomial for the codewords is given by

$$g(x) = LCM[f_1(x); f_2(x); ::::; f_{2t}(x)]$$

where $f_i(x)$ is the minimal polynomial of the i^{th} element of GF(p^m). Com-pute the value of k = n deg(g(x)). Maximum number of nodes in the network will be p^k. The value of t is chosen in such a way that p^k will cover the network size.

Step 5 : To obtain the polynomial of individual codewords, multiply each of the p^k number of node polynomials of degree k-1 in GF(p) with the generator poly-nomial. Here, each p^k number of node polynomials means all the polynomials of degree k-1 whose coe cients are from GF(p).

B. Second Phase

In this phase we derive the key identifiers for each sensor from the codewords formed in first phase. The codeword for each node derived in the Fifirst phase is mapped to key identifiers, which will identify the keys to be assigned to the sensor node. There is a unique key corresponding to each key identifier. We derive n key identifiers from codeword (a_1; a_2; a_3; ::::; a_n) where each identifier corresponds to an alphabet a_j for 1 j n. Each key identifier is a triplet, con-sisting of (a_j, j, s) where j = 1,2,.....,n and s is the relative position of appearance of the alphabet a_j in the codeword, i.e., s = (number of time a_j occurred in the codeword before the current occurrence + 1) mod $p^{m\ 1}$.

C. Example

We illustrate below the generation of BCH code and its mapping to key identifiers through an example.

©IJRASET: All Rights are Reserved

D. First Phase

Step 1 : Consider p = 2 and m = 3. The value of n is computed to be 7.

Step 2 : There are two primitive polynomials over GF(2) of degree 3. One is P (z) = z^3 +z+1 and another is P (z) = z^3 +z^2 +1. We randomly choose one primitive polynomial. In this example we consider the polynomial P (z) = z^3 + z + 1. Then we construct GF(2^3) as follows :

Step 3 : The conjugate sets and their corresponding minimal polynomials are given in Table 1

TABLE 1 CONJUGATE SETS AND THEIR CORRESPONDING MINIMAL POLYNOMIALS

Conjugate sets	Minimal polynomial
$\beta^1, \beta^2, \beta^4$	$(x^3 + x + 1)$
$\beta^3, \beta^6, \beta^5 (= \beta^{12})$	$(x^3 + x^2 + 1)$
β^7	$(x - 1)$

Step 4 : We consider the value of t = 1. The generator polynomial g(x) = LCM[(minimal polynomial of ^{1}) , (minimal polynomial of ^{2})]

= LCM[(x^3 + x + 1); (x^3 + x + 1)]

= (x^3 + x + 1).

Value of k = 7 3 = 4. Therefore, the number of nodes in the network is $p^k = 2^4 = 16$.

Step 5 : Each node have corresponding node polynomial of degree 3 in GF(2). We obtain the code polynomial for each node by multiplying the node polynomial for that node with the generator polynomial x^3 + x + 1. Codeword for each node is obtained from their respective code polynomial. Node ID, node polynomial, code polynomial and their corresponding codeword for the Sixteen nodes that we have considered in our example is shown in Table .

Second Phase : In this phase we derive the key chain for a node from its codeword. For example the key identifiers corresponding to Node ID 1 is shown in Table 1.

TABLE II

NODE ID ALONG WITH ITS CORRESPONDING NODE POLYNOMIAL, CODE POLYNOMIAL AND CODEWORD FOR SIXTEEN NUMBER OF NODES

Node ID	Node Polynomial	Code Polynomial	Code Representation
0	0	0	0000000
1	1	$x^3 + x + 1$	0001011
2	x	$x^4 + x^2 + x$	0010110
3	x + 1	$x^4 + x^3 + x^2 + 1$	0011101
4	x^2	$x^5 + x^3 + x^2$	0101100
5	x^2 + 1	$x^5 + x^2 + x + 1$	0100111
6	x^2 + x	$x^5 + x^4 + x^3 + x$	0111010
7	x^2 + x + 1	$x^5 + x^4 + 1$	0110001
8	x^3	$x^6 + x^4 + x^3$	1011000
9	x^3 + 1	$x^6 + x^4 + x + 1$	1010011
10	x^3 + x	$x^6 + x^3 + x^2 + x$	1001110
11	x^3 + x + 1	$x^6 + x^2 + 1$	1000101
12	$x^3 + x^2$	$x^6 + x^5 + x^4 + x^2$	1110100
13	$x^3 + x^2$ + 1	$x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$	1111111
14	$x^3 + x^2$ + x	$x^6 + x^5 + x$	1100010
15	$x^3 + x^2$ + x + 1	$x^6 + x^5 + x^3 + 1$	1101001

©IJRASET: All Rights are Reserved

TABLE III
KEY IDENTIFIERS FOR NODE ID 1

Codeword	0	0	0	1	0	1	1
Key identifiers	(0,1,1)	(0,2,2)	(0,3,3)	(1,4,1)	(0,5,0)	(1,6,2)	(1,7,3)

After obtaining the key identifiers for a node the keys corresponding to the key identifiers are installed in the node before deployment. Key identifiers of all the Sixteen nodes considered in our example are shown in Table II. Shared Key Discovery and Path-Key Establishment Phase

In the shared key discovery phase, every node will broadcast their key identifiers to all its neighbor nodes. Each neighbor nodes, after getting the key identifiers of a particular node will match with its own key identifiers. The keys corresponding to the matched key identifier will be the shared key between those two nodes.

For example Node 11 and Node 12 have a common key identifier (1,1,1). The key corresponding to the key identifier (1,1,1) is used as the shared key for the communication between them.

After the shared key discovery phase, if two neighboring nodes nd no shared key between them, then they will nd a third node who is connected to both the nodes and will establish the path key between the two nodes. For example, let A and B be the first two nodes and C be the third node. If k_1 is the key between A and C and k_2 is the key between B and C, then C will send k_1 key to B encrypting it with k_2, and will send k_2 key to A encrypting it with k_1. Both node A and B will create a secret key $K_{AB} = h(k_1; k_2)$ and erase k_1 and k_2 from their memory. The key K_{AB} is used as the secret key between A and B.

TABLE IV
KEY IDENTIFIERS OF ALL THE SIXTEEN NODES CONSIDERED IN THE EXAMPLE

Node ID	Key Identifiers
0	(0,1,1),(0,2,2),(0,3,3),(0,4,0),(0,5,1),(0,6,2),(0,7,3)
1	(0,1,1),(0,2,2),(0,3,3),(1,4,1),(0,5,0),(1,6,2),(1,7,3)
2	(0,1,1),(0,2,2),(1,3,1),(0,4,3),(1,5,2),(1,6,3),(0,7,0)
3	(0,1,1),(0,2,2),(1,3,1),(1,4,2),(1,5,3),(0,6,3),(1,7,0)
4	(0,1,1),(1,2,1),(0,3,2),(1,4,2),(1,5,3),(0,6,3),(0,7,0)
5	(0,1,1),(1,2,1),(0,3,2),(0,4,3),(1,5,2),(1,6,3),(1,7,0)
6	(0,1,1),(1,2,1),(1,3,2),(1,4,3),(0,5,2),(1,6,0),(0,7,3)
7	(0,1,1),(1,2,1),(1,3,2),(0,4,2),(0,5,3),(0,6,0),(1,7,3)
8	(1,1,1),(0,2,1),(1,3,2),(1,4,3),(0,5,2),(0,6,3),(0,7,0)
9	(1,1,1),(0,2,1),(1,3,2),(0,4,2),(0,5,3),(1,6,3),(1,7,0)
10	(1,1,1),(0,2,1),(0,3,2),(1,4,2),(1,5,3),(1,6,0),(0,7,3)
11	(1,1,1),(0,2,1),(0,3,2),(0,4,3),(1,5,2),(0,6,0),(1,7,3)
12	(1,1,1),(1,2,2),(1,3,3),(0,4,1),(1,5,0),(0,6,2),(0,7,3)
13	(1,1,1),(1,2,2),(1,3,3),(1,4,0),(1,5,1),(1,6,2),(1,7,3)
14	(1,1,1),(1,2,2),(0,3,1),(0,4,2),(0,5,3),(1,6,3),(0,7,0)
15	(1,1,1),(1,2,2),(0,3,1),(1,4,3),(0,5,2),(0,6,3),(1,7,0)

TABLE V
KEY IDENTIFIERS OF ALL THE NEWLY ADDED SIXTEEN NUMBER OF NODES IN THE NETWO

Node ID	Node Polynomial	Codeword Polynomial	Code after circular left shift
16	x^4 + 0	$x^7 + x^5 + x^4$	01100001
17	x^4 + 1	$x^7 + x^5 + x^4 + x^3 + x + 1$	01110111

©IJRASET: All Rights are Reserved

18	$x^4 + x$	$x^7 + x^5 + x^2 + x$	01001101
19	$x^4 + x + 1$	$x^7 + x^5 + x^3 + x^2 + 1$	01011011
20	$x^4 + x^2$	$x^7 + x^4 + x^3 + x^2$	00111001
21	$x^4 + x^2 + 1$	$x^7 + x^4 + x^2 + x + 1$	00101111
22	$x^4 + x^2 + x$	$x^7 + x^3 + x$	00010101
23	$x^4 + x^2 + x + 1$	$x^7 + 1$	00000011
24	$x^4 + x^3$	$x^7 + x^6 + x^5 + x^3$	11010001
25	$x^4 + x^3 + 1$	$x^7 + x^6 + x^5 + x + 1$	11000111
26	$x^4 + x^3 + x$	$x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x$	11111101
27	$x^4 + x^3 + x + 1$	$x^7 + x^6 + x^5 + x^4 + x^2 + 1$	11101011
28	$x^4 + x^3 + x^2$	$x^7 + x^6 + x^2$	10001001
29	$x^4 + x^3 + x^2 + 1$	$x^7 + x^6 + x^3 + x^2 + x + 1$	10011111
30	$x^4 + x^3 + x^2 + x$	$x^7 + x^6 + x^4 + x$	10100101
31	$x^4 + x^3 + x^2 + x + 1$	$x^7 + x^6 + x^4 + x^3 + 1$	10110011

IV. SCALABILITY OF THE SCHEME

The number of nodes N, that is addressable in the proposed scheme is p^k. For ad-dition of nodes into the network, we need to generate the codeword for the nodes to be added. To generate the codeword for the nodes to be added, all the k degree polynomials in GF(p) are multiplied with the generator polynomial, g(x). All the polynomials of k degree whose coefficients are from GF(p) are taken into account except those polynomials whose coefficient of x^k is zero. We are not considering polynomials whose coefficientof x^k is zero because a polynomial of degree k in GF(p) whose coefficientof x^k is zero is nothing but a k 1 degree polynomial in GF(p). The codeword for the new nodes will be of n + 1 bits. However, the codeword for the existing nodes are of n bits. We perform circular right shift of each newly formed codeword to get the desired codeword so that we can match the key identifiers for the same power of x in the code polynomials for any two nodes. The number of new nodes that can be added into the network is p^{k+1} p^k. The number of keys to be installed in the new nodes will be n+1, whereas the number of keys in the existing nodes will remain at n. We can add new nodes into the net-work without changing the keys of the existing nodes. Thus the scheme is scalable.

We explain the scalability by means of an example. Suppose a network has Sixteen number of nodes and the key identifiers associated with each node in the network is shown in Table V. Number of keys installed at each node is Seven. Suppose we want to add sixteen more nodes to the existing network.

Suppose an existing node, say Node 11, wants to communicate with a newly added node, say Node 27. Fifirst they will exchange their key identifiers. The key identifier (1,1,1) is found to be common between them. So, they will use the key corresponding to the key identifier (1,1,1) for secure communication between them. If no common key exists between them then they will create a secret key between them during the path-key establishment phase

V. RESULTS AND DISCUSSION

In this section, we have shown the result of network resiliency and number of keys per node for different values of number of nodes in the network. Here, we have assumed the value of m to 2. We have compared our result with two existing schemes , Camtepe and Yener scheme and Ruj and Roy scheme. The met-rics used for comparison is the number of node, the number of keys required per node, number of compromised nodes, and the resiliency.

We have shown the resiliency of our scheme against random node capture attack. It can be seen from Table 4 that our proposed scheme is more resilient.

©IJRASET: All Rights are Reserved

TABLE VI
COMPARISON OF PROPOSED KEY PRE-DISTRIBUTION SCHEME WITH RUJ AND ROY (R R) SCHEME AND CAMTEPE AND YENER (C Y) SCHEME. NUMBER OF NODES IN THE NETWORK IS N, KEYS PER NODE IS K, NUMBER OF COMPROMISED NODE IS S AND RESILIENCY IS FAIL(S)[FAIL(S) IS THE PROBABILITY OF A ECTED LINKS DUE TO THE COMPROMISE OF S NODES].

Proposed Scheme				R R Scheme				C Y Scheme			
k	N	s	Fail(s)	k	N	s	Fail(s)	k	N	s	Fail(s)
24	625	5	0.175251	22	529	5	0.217391	24	553	5	0.198915
24	625	10	0.316493	22	529	10	0.434783	24	553	10	0.397830
48	2401	10	0.142349	48	2401	10	0.186564	48	2257	10	0.203810
48	2401	15	0.229743	48	2401	15	0.2761	48	2257	15	0.305715
48	2401	20	0.355067	48	2401	20	0.408163	48	2257	20	0.407621
80	6561	10	0.070326	80	6561	10	0.123456	80	6321	10	0.123398
80	6561	20	0.230650	80	6561	20	0.246913	80	6321	20	0.246796

The deterministic schemes mentioned in our scheme is scalable. In our scheme, scalability can be increased without changing the keys of existing nodes
of the network

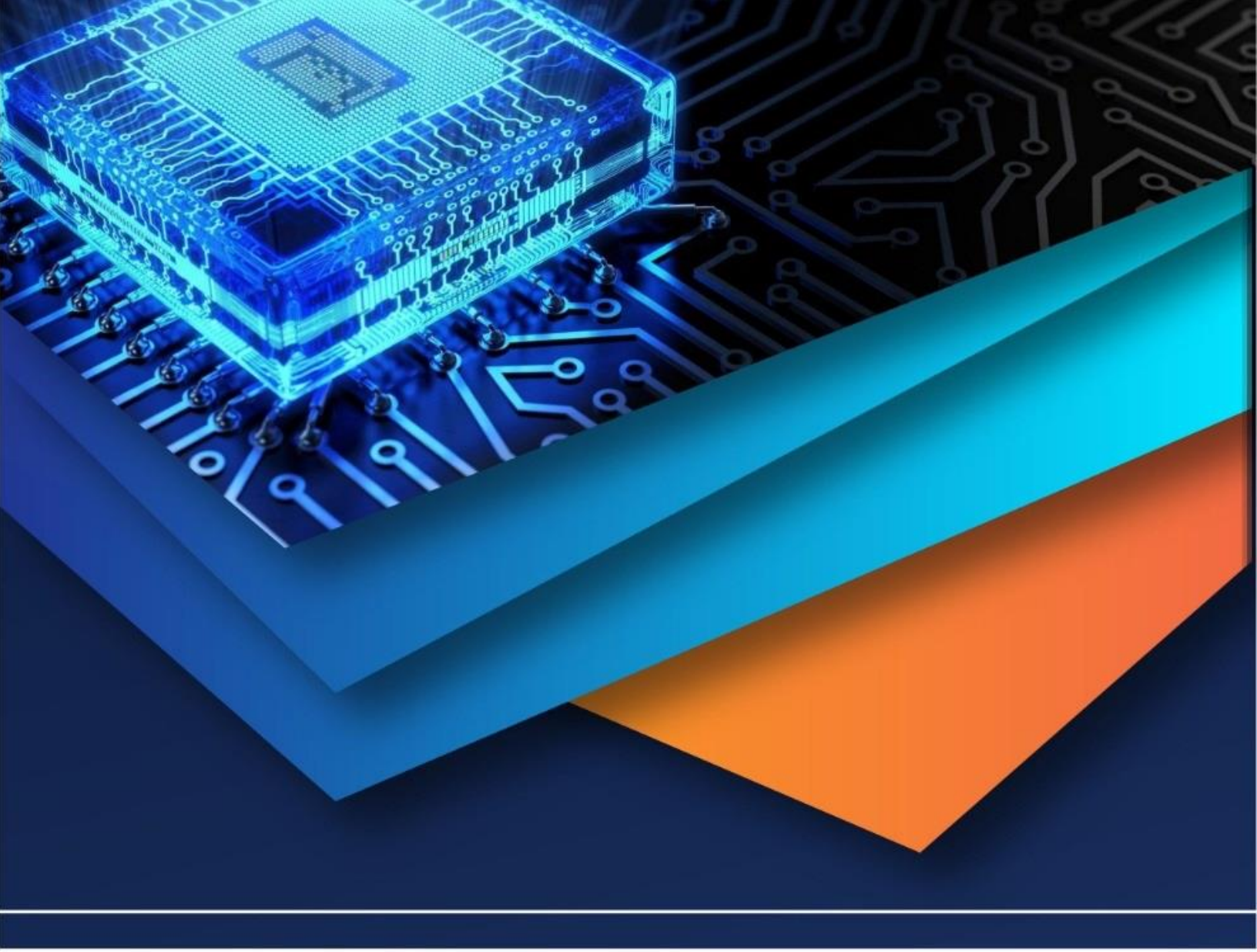
VI. CONCLUSION

In this work, we have proposed a key pre-distribution scheme using BCH codes. In the proposed scheme, we have taken key pool based approach where key identifiers of each node will be taken from a pool of key identifiers. The advantage of the scheme over the other deterministic schemes is that in this scheme, new nodes can be added to the network without changing the con guration of keys of the existing nodes. Also by varying the value of the different parameters in the scheme we can setup different sizes of network based on the requirement. Varying the values of t, which is the number of errors BCH code can correct, we can accommodate desired number of nodes with some acceptable resiliency. In future we would like to use other coding scheme and see their performance.

REFERENCES

[1] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In CCS '02: Proceedings of the 9th ACM con-ference on Computer and communications security, pages 41-47, New York, NY, USA, 2002. ACM.
[2] Seyit Ahmet Camtepe and B•ulent Yener. Combinatorial design of key distri-bution mechanisms for wireless sensor networks. In ESORICS, pages 293-308, 2004.
[3] Seyit A. Camtepe and B•ulent Yener. Combinatorial design of key distri-bution mechanisms for wireless sensor networks. IEEE/ACM Trans. Netw., 15(2):346{358, 2007.
[4] Jooyoung Lee; D.R. Stinson. A combinatorial approach to key predistribution for distributed sensor networks,. Wireless Communications and Networking Conference, 2:1200{1205, 13-17 March 2005.
[5] Jooyoung Lee; D.R. Stinson. Common intersection designs,. Journal of Com-binatorial Designs, 14:251{269, 2006.
[6] Sushmita Ruj and Bimal Roy. Key predistribution schemes using codes in wireless sensor networks. pages 275{288, Berlin, Heidelberg, 2009. Springer-Verlag.
[7] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In SP '03: Proceedings of the 2003 IEEE Sympo-sium on Security and Privacy, page 197, Washington, DC, USA, 2003. IEEE Computer SocietyHaowen
[8] .Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In SP '03: Proceedings of the 2003 IEEE Sympo-sium on Security and Privacy, page 197, Washington, DC, USA, 2003. IEEE Computer Society
[9] .Donggang Liu and Peng Ning. Establishing pairwise keys in distributed sensor networks. In ACM Conference on Computer and Communications Security, pages 52-61, 2003.
[10] .Carlo Blundo, Alfredo De Santis, Amir Herzberg, Shay Kutten, Ugo Vaccaro, and Moti Yung. Perfectly-secure key distribution for dynamic conferences. In CRYPTO, pages 471{486, 1992.
[11] .Donggang Liu, Peng Ning, and Rongfang Li. Establishing pairwise keys in distributed sensor networks. ACM Trans. Inf. Syst. Secur., 8(1):41-77, 2005.
[12] Sencun Zhu, Shouhuai Xu, Sanjeev Setia, and Sushil Jajodia. Establishing pairwise keys for secure communication in ad hoc networks: A probabilistic approach. In ICNP, pages 326-335, 2003.
[13] Donggang Liu, Peng Ning, and Wenliang Du. Group-based key predistribu-tion for wireless sensor networks. TOSN, 4(2), 2008.
[14] Haowen Chan, Virgil D. Gligor, Adrian Perrig, and Gautam Muralidharan. On the distribution and revocation of cryptographic keys in sensor networks. IEEE Trans. Dependable Sec. Comput., 2(3):233-247, 2005.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)