KCSR: KEYMATCHES CONSTRAINED SECURE ROUTING IN HETEROGENEOUS WIRELESS SENSOR NETWORKS

K. Shaila, G. H. Vineet, C. R. Prashanth, V. Tejaswi, K. R. Venugopal
Department of Computer Science and Engineering, University Visvesvaraya College of Engineering
Bangalore University, Bangalore 560 001, India

L. M. Patnaik
Vice Chancellor, Defence Institute of Advanced Technology, Pune, India

Keywords: Heterogeneous Networks, Key Distribution, Node Compromise, WSNs Security.

Abstract: Wireless Sensor Networks (WSNs) consists of a large number of tiny autonomous devices called sensors which are vulnerable to security threats. Heterogeneous Wireless Sensor Networks consists of two types of nodes L1 and L2 that employees Unbalanced Key Distribution Scheme to ensure enhanced security. In this paper, the concept of *Link Strength* is utilized, which determines the path to be selected for secure routing. Our Keymatches Constrained Secure Routing(*KCSR*) algorithm provides the flexibility to choose secure path and then route the data accordingly. In this approach, secure and stable paths are chosen for communication. Simulation results show that the proposed algorithm yields better results emphasizing security when compared with earlier works.

1 INTRODUCTION

Wireless Sensor Networks (WSNs) consists of spatially distributed autonomous devices. They are used to assimilate and interpret real time data in smart environment applications like in military bases or vehicle target tracking systems. Owing to the criticality of data, many applications require secure communication. WSNs are vulnerable to security attacks: due to the broadcast nature of transmission, easily accessible to the attackers and the nodes are exposed which can be destroyed.

WSNs needs a secure key management technique to protect itself from any attack that targets confidentiality, integrity and authentication properties of its communication channels. A well appreciated solution that has been widely used is random deployment of keys in a balanced network. Such a homogeneous network made the nodes more vulnerable to security breaches.

Motivation: Predeployed nodes exchange encrypted messages to establish communication. Secure communication is established depending on the number of keymatches. Multiple common keys provides enhanced security compared to a single key match. The security level is enhanced by considering a heterogeneous network with an asymmetric key management technique.

Contribution: We utilize an asymmetric predistribution of keys for all nodes in heterogeneous environment. Nodes that desire to communicate, exchange their key identifiers along with the keys to find a match. Instances wherein there are no key matches, leverages the use of small percentage of more capable sensor nodes with enhanced level of security. We have proposed *Link Strength* that depends on the number of key matches and total number of secure links as a measure of security in the network. When the keys between any two nodes match with each other, then a link is established between the nodes.

Organization: The remaining part of the paper consists of a brief review of Related works in Section 2. Section 3 explains System Model and Problem Definition. A mathematical model is derived in Section 4 followed by Algorithm in Section 5. Section 6 analyses the performance and conclusions in Section 7.

Shaila K., Vineet G., Prashanth C., Tejaswi V., Venugopal K. and Patnaik L..
KCSR: KEYMATCHES CONSTRAINED SECURE ROUTING IN HETEROGENEOUS WIRELESS SENSOR NETWORKS.
DOI: 10.5220/0003438700130022
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 13-22
ISBN: 978-989-8425-56-0

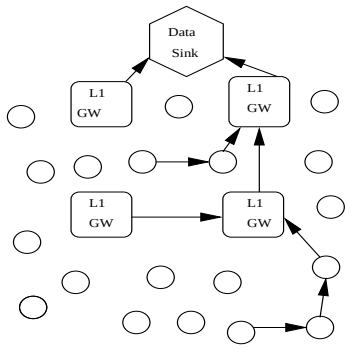

Figure 1: Model for Heterogeneous networks consisting of two types of nodes.

2 RELATED WORK

WSNs are small autonomous networked, low power bodies, called *motes* and are made of piezoelectric material (F. L. Lewis and Wiley, 2004). WSNs consists of spatially distributed autonomous devices used for distributed information accumulation in unattended areas. Eschenauer et al., (Eschaenauer and Gligor, 2002) proposed a Probabilistic Key Predistribution Scheme for pairwise key establishment. Each senor node randomly picks a set of keys from a key pool before deployment so that any two sensor nodes have a certain probability of sharing atleast one common key.

Du et al., (Wenliang Du and V, 2003) proposed a new predistribution scheme which uses pairwise keys that enable authentication. It assures substantially improved network resilience against node capture over the existing schemes. Chan et al., (Haowen Chan and Song, 2006) further extended this idea and developed two key predistribution techniques: *q* composite key predistribution and random pair-wise key distribution scheme. This scheme randomly picks pairs of sensors and assigns each, a pair of unique random keys.

Traynor et al., (Traynor et al., 2006)(Traynor et al., 2007) proposed LIGER, a Hybrid Key Management Scheme with the presence and absence of a Key Distribution Center (KDC). When KDC is not available, nodes communicate securely with each other based upon a probabilistic unbalanced method of key management. They probabilistically authenticate neighboring devices with which they are communicating.

Perrig et al., (Perrig et al., 2002) proposed Security Protocols for Sensor Networks(SPINS). Here, each sensor node shares a secret key with the base station. Debao et al., (Debao Xiao and Zhou, 2006) extended SPINS to provide additional security. Jian Chao et al., (Chao and Xiuli, 2008) presents an overview of the various attacks and defenses on each of the concerned layers in Sensor Networks. Roman *et al.*, (Rodrigo Roman and Lopez, 2005) identified three basic factors in the design of a key infrastructure for WSNs: *Key Storage*, *Key Distribution* and *Key Maintenance*.

Yong Wang et al., (Wang and Ramamurthy, 2008) use five different types of keys to ensure security in the Sensor Network. The protocol assumes the compromise of the base station and sensor nodes. Karlof et al., (Chris Karlof and Wagner, 2004) proposed a link layer security architecture for WSNs, where, the keying mechanism described gives an emulated overview about *Tinyseckey*.

Wang et al., (Yong Wang and Xue, 2008) proposed security of the base station and proposes *m* keying which has two schemes namely, key predistribution scheme and key revocation scheme. Chao *et al.*, (Chao and Xiuli, 2008) presents an overview of the various attacks and defenses on each of the concerned layers in Sensor Networks. A number of key distribution schemes exists, but the most widely used is the *q*-composite Key Predistribution Scheme. A matrix key distribution presents a novel bidirectional method which generates a number of combinations. However, the computation increases drastically consuming a large amount of energy, reducing the efficiency of WSNs.

Boujelben *et al.*, (Manel Boujelben and Youssef, 2009) have proposed a pairwise Key Management protocol that is applied to two tiered Heterogeneous Wireless Sensor Networks. They have proposed probabilistic key predistribution which is used in the lower tier of the network architecture and public key cryptography in the upper tier. Our scheme provides a paradigm to provide security through *Link Strength* of the path.

3 MODEL AND PROBLEM DEFINITION

3.1 System Model

The sensor network consists of static heterogeneous nodes as shown in the Figure 1, with Unbalanced Key Distribution and provides secure hop-to-hop communication. It minimizes the burden of less capable nodes and routes the encrypted data through more capable nodes. Enhancement of security is an added advantage when Unbalanced Key Distribution is incorporated in Heterogeneous Network Scheme. We have considered two types of nodes, more capable nodes as Level1(L1) nodes and less capable nodes are repre-

sented with circles as Level2(L2) nodes. The number of L1 nodes is (1/8)th *i.e.*, 12.5% of the entire network(Traynor et al., 2006). The effect of node compromise is less when L1 nodes are (1/8)th of the network.

The network communication is established by exchange of keys among the nodes. Communicating nodes sharing a common key become trusted neighbors. However, malicious nodes could manipulate messages during transmission. This leads to security breach. Depending on the data to be routed, it would be appropriate to surpass the network traffic through L1 nodes, which inturn reduces the burden on L2 nodes, providing increased level of security and minimizing the effects of node compromise.

3.2 Problem Definition

WSNs are vulnerable to attacks due to broadcast nature of transmission medium, resource limitation on sensor nodes and uncontrolled environments where they are left unattended. The chances of threats and attacks are common with Homogeneous Networks. Such risks can be minimized using the Heterogeneous Networks. The main objectives of this work is to:

(i) Find maximum number of key matches between the communicating nodes.

(ii) To select the path with higher degree of security.

3.3 Assumptions

In the heterogeneous model,

(i) L1 nodes are robust and have high processing capability. Based on the shared key, they take the role of gateway in the network.

(ii) In addition to tamper resistant casings (F. L. Lewis and Wiley, 2004)(Traynor et al., 2007), L1 nodes are assumed to be equipped with fast encryption and decryption algorithms to protect their additional keys from compromising if captured. Chances of L1 nodes getting compromised/captured is minimum.

(iii) L2 nodes have less sensing capacity, memory and number of keys deployed. L2 nodes sense and accumulate the data.

Link Strength for a given node in the network is defined as the ratio of total number of keymatches to the total number of links. Based on the value of *Link Strength*, an appropriate path(direct or indirect) is chosen to route the data securely for a given application.

$$Link\ Strength = \frac{\sum_{i=1}^{n} X_i}{L} \tag{1}$$

X_i = Common keys in the direct and indirect path, where i varies from 1 keymatch to n keymatches and L is the total number of links amongst the communicating nodes in the network.

4 MATHEMATICAL MODEL

(i) $THEOREM$: In a Heterogeneous Wireless Sensor Network. Let N be the total number of nodes consisting of L1 and L2, with 12.5% of N being L1 based on capability, cost, efficiency and optimization of security. Consider, a global key pool of size P. Let m keys be deployed in L1 nodes and k keys be deployed in L2 nodes such that $P>>m>>k$.
(ii) *Statement* : The *Link Strength* varies exponentially with the threshold keys or number of common keys after the direct phase.

$$Link\ Strength \propto e^{T_i}$$

where, T_i is the threshold keys or number of key matches after direct phase and i varies from *1,2,3. . . n*.

(iii) $Proof$: Consider the predeployment of m keys in L1 nodes without replacement. Number of ways in choosing m keys from global key pool P is,

$$\binom{P}{m} = \frac{P!}{m!(P-m)!} \tag{2}$$

Similarly, deploy k keys from $(P-m)$ keys in L2 nodes. Number of ways in choosing k keys from $(P-m)$ is,

$$\binom{P-m}{k} = \frac{(P-m)!}{k!(P-m-k)!} \tag{3}$$

The probability(M_i) of not finding atleast one key match is determined by the ratio of number of ways of choosing keys for L1 followed by L2 to total connections possible $\binom{P}{k}$ which is given using Eq. 2 and Eq. 3.

$$M_i = \frac{(P-m)!(P-k)!}{P!(P-m-k)!} \tag{4}$$

The number of keymatches between any two communicating nodes after direct or indirect keymatch phase is represented in Eq. 4. Thus, the probability of atleast one keymatch (M_i) is,

$$M_i = 1 - \frac{(P-m)!(P-k)!}{P!(P-m-k)!} \tag{5}$$

Next, the probability of exactly one key match is,

$$M_1 = 1 - \frac{k(P-m)!(P-k)!\binom{m}{1}}{P!(P-m-k+1)!} \tag{6}$$

Table 1: Notations.

Symbols	*Definition*
dst	Destination Node
src	Source Node
$Rnd()$	Pseudo Random Generating Function
K_r	Maximum possible value that is generated
n_{gk}	Maximum global keys list size
n_{lk1}	Maximum number of keys present in $L1$ node
n_{lk2}	Maximum number of keys present in $L2$ node
$GK[1-2][1...n_{gk}]$	List of global keys
$LK1[1-2][1...n_{lk1}]$	List of local keys present in $L1$ nodes
$LK2[1-2][1...n_{lk2}]$	List of local keys present in $L2$ nodes
N	Total number of nodes in the network
T_i	Number of common key matches after direct phase
L	Total number of links in the network
DC_{mk}	Common key matches through direct phase
IDC_{mk}	Common key matches through indirect phase
$Exch[1..]$	List of exchangeable key identifiers
$L2Neigh[j]$	Gateway node $L2$ with node j
$Neigh[j]$	List of Next Hop node

Deploy one key from a pool of m keys to L1 ($m \subseteq$ P). Using Eq. 4 and Eq. 5, the probability of exactly x keymatches is derived as to be:

$$CM_x = \left(\frac{k(P-m)!(P-k)!\binom{m}{x}}{P!(P-m-k+x)!} \right) \tag{7}$$

where, x = 1, 2, . . . n.

Total links in the network consists of direct links and indirect links. Total connectivity in the network consisting of N nodes is given by,

$$Total\ connectivity = N(N\text{-}1).$$

However, the total connectivity equation holds true if the connection is one-to-one and ideal. In the real scenario, the network connectivity may not be the same. Hence, derive the probability of having indirect links in the network. Probability of having no direct links (P_{nd}) or indirect links is defined as summation of the probability of having exactly i keymatches varying over a threshold T_1 to T_n. Substituting for x = 1, 2, . . .,n in Eq. 7, P_nd is,

$$P_{nd} = \left(\frac{k(P-m)!(P-k)!}{P!} \right) \left(\sum_{i=1}^{n} \frac{\binom{m}{i}}{(P-m-k+i)!} \right)$$

$$= \left(\frac{k(P-m)!(P-k)!(2^m-1)}{P!} \right) \left(\sum_{i=1}^{n} \frac{1}{(P-m-k+i)!} \right) \tag{8}$$

Since P is very large, we use Stirling's approximation for $n!$

$$n! \approx \sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n}$$

Finally, the expression obtained from Eq. 8 is,

$$P_{nd} = \left(\frac{k(P-m)!(P-k)!(2^m-1)}{P!\sqrt{2\pi}} \right) \left(\sum_{i=1}^{n} \frac{e^{\beta}}{\beta^{\beta+\frac{1}{2}}} \right) \tag{9}$$

where, β is $(P-m-k+i)!$

Illustration : The key factor to determine the degree of security while routing is *Link Strength*. Higher the value of Link Strength, more secure is the path for the data to be routed. In case of indirect communication between the nodes, the value of Link Strength is the average of common keys between L1 and L2. Since the value of Link Strength is dependent on the number of links, this is inturn a factor of number of nodes in the network. Network being heterogeneous, value of *Link Strength* is the total of the values in the direct and indirect communication.

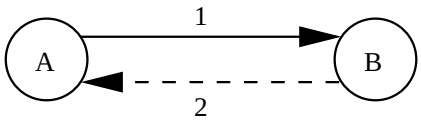

Figure 2: Stage 1: Establishment of direct path. Where, 1-Request sent for establishing direct path and 2-Key sent as an acknowledgment, assuring secure link.

The method described provides the flexibility to use any paths to route the data based on the degree of security required for a particular application. When different values of n are substituted in Eq. 9, we get different values of P_{nd}. This value is used to calculate T_i, where, i varies from $1, 2, ...n$. As an example, consider three common keys are required for secure communication. The value of T_3 is obtained by substituting 3 for i in Eq. 9. If this value suits for a given application, then three keymatches are chosen to route the data securely. Similarly, try for different values of i, *i.e.*, common keys matches. According to the theorem, as the value of T_i increases with

i, the *Link Strength* also increases exponentially and increases the degree of security to route the data.
However, if more number of common keys are chosen then, the data transfer becomes complicated and therefore time required is more. Some applications require the data to be routed quickly. In such a case, the number of common keys required for communication is decreased and the data is routed at higher speed. Thus, for any given application, the data can be routed with assured security. The value of *Link Strength* calculated becomes the theoretical value. However, this method of calculating *Link Strength* is computationally intensive since Eq.9 is complex.

5 ALGORITHM

A secured data routing must be established among the various paths available that jeopardize the effects of node compromise. Based on *Link Strength*, the proposed method passes the data through an indirect link, even though the direct link exists. This ensures higher degree of security to the network. The notations used in this paper are defined in Table 1. The various stages in the algorithm are:

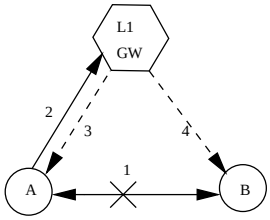

Figure 3: Stage 2: Establishment of indirect path. Where, 1-Direct path establishment is not possible. *X* indicates its failure, 2-Request sent to route data via L1(gateway) to B, if path is secure and 3,4-Acknowledgment given by L1 to request L2 nodes.

5.1 Unbalanced Key Distribution Phase

Unbalanced Key Distribution of keys depends on the previous approach proposed by Gligor (Eschaenauer and Gligor, 2002). In the balanced approach, out of the total *P* keys in the pool, if *k* are randomly selected without replacement, then the probability of the two nodes having same common keys *k*, sharing atleast one key is determined. However, if there exists no keymatch, an alternate path is established through one or more intermediate nodes having common keys. Given, the same key pool of size *P*, store a pool of keys of size *m* in L1 nodes and a key pool of size *k* in L2 nodes where $(m >> k)$.

An array of global key pool of size n_{gk} is generated. Local key pool is generated after the generation of global keys with unique key identifiers $LK1[1][1...n_{lk1}]$ or $LK2[1][1...n_{lk2}]$. All local keys are subset of global keys and $LK1[2][1...n_{lk2}||n_{lk2}]$=$GK[2][1...n_{gk}]$.

Once the keys are deployed both L1 and L2 nodes learns its neighbors through HELLO messages. In the given transmission range, each node broadcasts its key identifiers to its neighbors and finds the node(s) that share a common key, to communicate. The secure path is established if there exists a key match. However, the best secure path as per our scheme is not yet chosen. In this method, we transmit the key identifiers and not the key themselves, because sufficient information can be gathered as in traffic analysis attack (Haowen Chan and Song, 2006). Thus, the neighboring L2 node in response sends the key identifier to the source as an acknowledgment, thus making the link secure.

However, if no keymatch exists, then L2 nodes establishes the link *via* L1. Since L1 nodes have more keys, communication through L1 is safe and secure. The communication process is initiated, when L2 nodes wishing to communicate, sends its identifiers to L1. L1 node(s) checks with its own set of key identifiers. When there exists a keymatch between L1 node and set of L2 nodes, L1 node sends an acknowledgment message to the set of L2 nodes. Thus, communication *via* L1 ensures trusted path. Different stages in direct and indirect communication are shown in the Figure 2 and Figure 3.

5.2 Selection of Path based on *Link Strength*

The keymatch phase as described earlier, minimizes the effects of node compromise, but not to a larger extent. If the intruder overhears and successfully decrypts the shared key, the link becomes insecure. It would disrupt the network and sabotages the efficient secure link that was established initially. In the proposed method, *Link Strength* is a measure of degree of security and is a factor of number of common keys. Its value varies for direct and indirect communication. Security gets maximized if the number of common keys increases.

Although there exists direct path for a set of L2 nodes, depending on the type of data, situation, topology and application, encrypted data is communicated in an indirect way. To ensure enhanced security, data is routed *via* L1, since L1 has more keys. If the number of common keys are more, it becomes difficult for an intruder to decrypt all the common keys. The value of *Link Strength* is computed in both the

Table 2: KCSR Algorithm: Path Establishment.

```
Path_Establishment()
for i = 0 to n_gh do
    dst ← Neigh[i]
    src ← presentnode()
    for j = 0 to n_lk1 or n_lk2 do
        Exch[j] ← LK[1][j]
        j ← j+1
    end for
    i ← i+1
end for
for i = 0 to n_lk1 or n_lk2 do
    for j = 0 to n_lk1 or n_lk2 do
        if LK1[1][j]Exch[i] or LK2[1][j]Exch[i]
        then
            DC_mk ← DC_mk + 1
        else
            IDC_mk ← DC_mk[i] + DC_mk[j]
        end if
        KeyDC_mk[i] ← LK1[2][j]orLK2[2][j]
        j ← j+1
        i ← i+1
    end for
end for
```

Table 3: KCSR Algorithm: Selection of Path Phase.

```
Path_Selection_Phase()
j ← 0
for i = 0 to n do
    if DC_mk < IDC_mk then
        dst ← L2Neigh[j]
        src ← presentnode()
    else
        dst ← Neigh[j]
        src ← presentnode()
    end if
    for i = 0 to N do
        if DC_mk < L_th or IDC_mk < L_th then
            Link between node j and i is not se-
            cure
            if DC_mk = L_th or IDC_mk = L_th then
                Link between node j and i is
                moderately secure
            else
                Link between node j and i is very
                secure
            end if
        end if
        j ← j+1
        i ← i+1
    end for
end for
```


phase. After the direct communication phase, the number of common keys to establish communication and to route the data is increased. These values are used in computation of threshold *Link Strength* and are called as *threshold keys*. The probability of having common keys for nodes communicating *via* L1 is higher, when the common keys required for communication is not greater than threshold keys. The path thus selected (direct or indirect) would provide more resilience against compromise and active attacks.

Each node of the network explores the available resources in its vicinity, before getting exposed to any of the phases in the algorithm. Initially, before the deployment of nodes a global pool of keys have to be generated. This is done by using a random function, the seed value has to be chosen so that each value is unique. The size of global key chosen must be much higher than local keys deployed in each node $n_{gk} >> n_{lk1} >> n_{lk2}$.

After the global pool has been generated, random selection of keys and key identifiers from the pool are stored in the corresponding node. The node is then deployed in a grid location. The random selected keys should not exceed n_{lk2}. Randomness in the selection process determines a probability of match, which lies between 1 and 0. Suitable encryption and decryption algorithm, have to be implemented to ensure that the global key pool is resilient to any attack. The key pool may be dropped after deployment of nodes.

The proposed *KCSR* algorithm uses important features like unbalanced key distribution and predeployment of keys. The three distinct stages of the algorithm are:

(i) Direct Path Establishment Phase.
(ii) Indirect Path Establishment Phase.
(iii) Path Selection Phase.

(i) Direct Path Establishment Phase
Every node in the network sends a HELLO packet to all the adjacent nodes. The nodes in the neighborhood respond to the source by sending an ACK. ACK contains corresponding node ID and its grid positions. This information gets updated by the source node. Every node maintains a neighbor table, which includes various routing information. Two essential fields are the next hop and matched keys, which are widely used by the *KCSR* algorithm for secure routing. The unique key identifiers stored in the local key pool are broadcasted to the neighboring nodes. The recipient node verifies the set of key identifiers with its own set of key identifiers. The path establishment phase is illustrated in Table 2.

(ii) Indirect Path Establishment Phase
In case the direct path establishment fails due to lack of shared keys, the process of searching indirect path is initiated. Most of such failed nodes are usually observed to be L2 nodes, as the probability of a match is much lower when compared to L1 nodes. L2 nodes desiring to communicate, but having no common key sends their key identifiers to their neighboring L1 nodes. L1 nodes check the sets of identifiers sent with its own set. The common key identifier is sent

as a response indicating that it can play the role of a router/gateway to communicate between requesting L2 nodes, thus establishing an indirect link.

Owing to the fact that L1 nodes have more keys deployed in them, this path assures more security. Counters are set up at each node to keep track of the number of common key matches obtained. This phase may be used repetitively to find secure path between two nodes, which have low *Link Strength*.

(iii) Path Selection Phase
This phase is responsible for choosing the right path between a given source and destination node. The path chosen should have higher *Link Strength* and ensure secure communication. Based on this value, different links in the network are categorized as low, moderate and high secure paths. Depending on the requirements, selection of path is done and the data is routed accordingly. Path Selection Phase allows data to be routed at different levels of security or at the same level of security in the network for a given application. Path chosen once at a particular threshold does not imply that selection should always be made for the same threshold. Switching to different thresholds is also permissible making all nodes participate in the network. The path selection phase is given in Table 3.

5.3 An Example

In the proposed scheme, the steps followed to select the path are as follows:

(i) All the nodes during the direct path establishment are first identified and the value of *Link Strength* is determined.

(ii) *Link Strength* is calculated for nodes communicating in the indirect path.

In the first stage, direct communication is established if there exists one or more common keys(nodes are not communicating *via* L2). The constraint for establishing direct communication is atleast 1. At the last stage, after the direct path phase, we alter the constraint for direct communication by changing the common keys to be greater than 1, 2, 3 and so on. The path is selected by comparing the old and new values of *Link Strength* and selecting the one which has higher value. This process would assure resilience to the network against node compromise and defend itself from the active and passive attacks. The graphs for *Link Strength* for a given set of nodes *vs* the *key matches after the direct phase* or *threshold keys* is plotted and then analyzed.

Consider a network consisting of an L1 node and two L2 nodes. If the direct path is established between two L2 nodes the value of *Link Strength* depends on the number of keys deployed in L2. If *n* keys are deployed in L2 and all the keys are common, the maximum value of *Link Strength* is equal to (*n*/number of links). Let us consider an indirect path to be established inspite of a direct path. If there exists *n* common keys between L1 and L2, then *Link Strength* is determined to be as [(*n*1+*n*2)/2]. Both the direct and indirect path values are compared, it is observed that the indirect path is better than a direct path. But, having all the keys common is an ideal case.

The communication is open even if there exists a single key match. The security of open path depends on the number of common keys. Thus, if the threshold for the communication increases as 1, 2, . . . so on (common keys), then the value of *Link Strength* increases ensuring enhanced security. Owing to the fact that number of keys deployed in L1 is far greater than L2, the probability of finding common keys between L1 and L2 nodes is greater than finding common keys between two L2 nodes. This is not true for all instances. There may exist an indirect path, whose *Link Strength* is low compared to direct path. In such a case direct path is selected. Thus, based on the above comparisons, the paths are distinguished as low, moderate and high secure paths and the data is routed.

Successful active attacks allow the intruder to disrupt the functioning of the network. Attackers can masquerade the network by overhearing the messages and cause malfunctioning in the network. If an attacker compromises the node, then he can spy the network and gains full control over it. Such cases must be avoided and the network must ensure confidentiality and security. The analysis against secure threats is dependent on *Link Strength* which is defined as a function of number of common keys. It becomes more difficult for the intruder to decrypt all the keys that is open for communication with trusted neighbors. Moreover, the constraint for the number of common keys is again not disclosed. Therefore, the intruder is not aware as to how many common keys he needs to decrypt and the actual keys for communication. This ensures a double protected mechanism. Therefore, security is ensured in both the ways and proves to be more stable.

6 PERFORMANCE ANALYSIS

In order to implement the proposed Keymatches Constrained Secure Routing (*KCSR*) algorithm three messages are considered: *Send* message, *Send Path* message and *Update* message. These messages are de-

Table 4: Simulated values for a set of 10 nodes to determine Link Strength.

Key matches after direct phase (X)	Link Strength(y)
0	20.14856
1	23.90492
2	26.38660
3	27.22280
4	25.21774
5	26.95550

livered to specific nodes in the network at different phases of the algorithm. *Send* message is used to build a neighbor table at every node since the algorithm is executed in distributed fashion. After the first phase of the algorithm every node sends a *Send* message updating its status based on the common key. This updated information is further used by the algorithm to establish indirect links. *Send Path* message is given to the neighboring L1 nodes. Based on the previous status information the indirect path is established. *Update* message is used to cross verify the path chosen by a node. The next hop for the *Send* packets are updated. It is used in Path Selection Phase of the algorithm.The new *Link Strength* values are calculated. The path is selected by comparing the old and new values of *Link Strength* and selecting the one which has the larger value. This process would assure resilience to the network against node compromise and defend itself from the active and passive attacks. The graphs for *Link Strength* for a given set of nodes versus the *key matches after the direct phase* or *threshold keys* is plotted and then analyzed.

The graph is plotted for the value of *Link Strength* versus *key matches after the direct phase* for a given set of nodes in the network. The nature of the curve is determined by the numerical and Mathematical analysis. The process of finding the curve of best fit is called as *curve fitting*. The method of *least squares* is employed in our scheme for curve fitting. The curve analysis results in proper setting of number of common keys to achieve optimum security. Based on exponential behavior of the curve, as shown in Eq. 9 we can fit the curves of the form $y = ab^x$ where, a, b are the constants to be determined for the given set of x and y points. Therefore, the given curve takes the form,

$$Link\ Strength = ab^{T_i} \tag{10}$$

where, T_i is number of keymatches after the direct phase. *i.e.*, taking natural logarithms on both sides,

Table 5: Computation of X and Y using Least Square Method.

x=X	y	Y=ln y	XY	X^2
0	20.14856	3.00313	0	0
1	23.90492	3.17408	03.17408	1
2	26.38660	3.27285	06.54510	4
3	27.22280	3.30405	09.91215	9
4	25.21774	3.22754	12.91016	16
5	26.95550	3.29418	16.47090	25

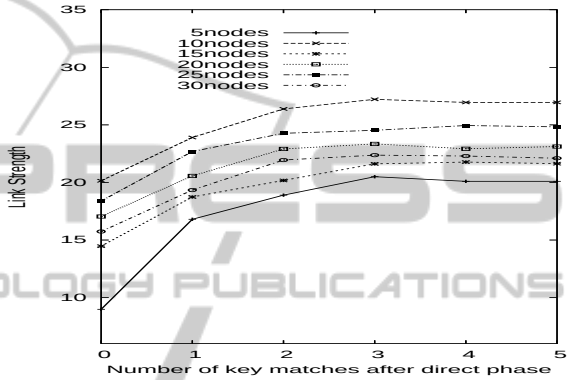

Figure 4: Analysis of *Link Strength* for different nodes in the network (simulation).

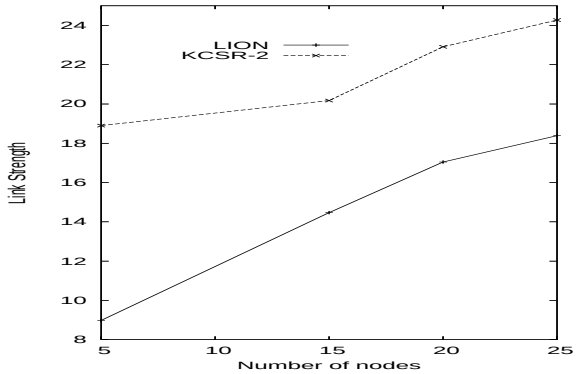

Figure 5: Improvement in *Link Strength* vs number of nodes.

we get

$$logy = loga + xlogb \tag{11}$$

or

$$u = A + Bx \tag{12}$$

where, A = log a and B = log b. Solving Eq. 11 and Eq. 12 the normal equations that yield A and B are

$$\sum u_i = nA + B \tag{13}$$

$$\sum x_i u_i = A\sum x_i + B\sum x_i^2 \quad (14)$$

where, u_i = log y_i. Consider the simulated values for 10 nodes as shown in Table 4. The zero value retains the same value of *Link Strength* as in direct communication whereas, the other values shows the new *Link Strength* value computed in indirect communication. The values $\sum X, \sum Y, \sum XY, \sum X^2$ are determined from Table 5 by using *method of least squares*. Substituting the above computed values in Eq. 13 and Eq. 14, the values of A and B are 3.095007 and 0.047513 respectively. The values of a and b are 22.08740 and 1.04866 respectively using, $a = e^A$ and $b = e^B$. The *Link Strength* obtained is

$$Link\ Strength = (22.08740)(1.04866)^{T_i} \quad (15)$$

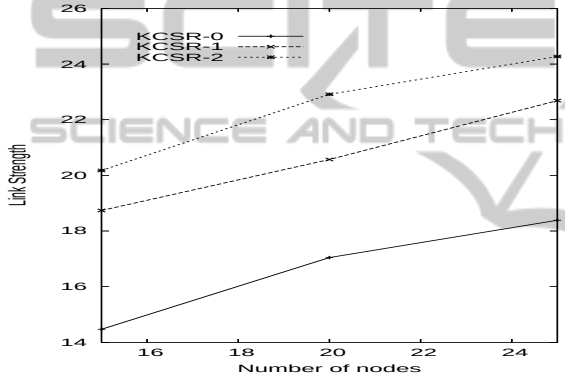

Figure 6: Variation in *Link Strength* for different key matches in the network.

where, T_i is the number of i key matches after direct phase in which i varies from *0, 1, 2, . . . 5*. Computations are done for 5, 15, 20, 25 and 30 nodes.

6.1 Simulation Setup

The simulation is performed in NS-2.31 simulator. The topological area is considered to be 50*m*x50*m*. The transmission range is set to 50*m*. A set of nodes with 12.5 percent of the entire network being L1 is deployed as in (Traynor et al., 2006)(Traynor et al., 2007). The deployment is random and uniformly distributed. L1 nodes are assumed to have more energy and processing capabilities than L2.

6.2 Simulation Results

The graph is plotted for the value of *Link Strength vs key matches after the direct phase* for a given set of nodes in the network. The behavior of curve analysis show that *Link Strength* varies exponentially with the *threshold key*. The variation in *Link Strength* for different number of key matches which are also known as *threshold keys* for different nodes in the network is shown in Figure 4. The value of *Link Strength* increases for different nodes in the network. When the number of key matches increases, *Link Strength* also increases, but becomes constant after three key matches. This shows the exponential behavior of *Link Strength*.

In Figure 5, the values of *Link Strength* for different nodes in the network is analysed. The two results indicate that one is for zero key matches (LION)(Traynor et al., 2007) and the other is for two key matches (*KCSR*). An improvement of 60%-70% on an average, exists in *Link Strength* value. The proposed scheme yields 65% better results as compared to other scheme. The results indicate that the path selected is 65% more secure.

The results of *Link Strength* for a variable number of common keys is shown in Figure 6. As the number of nodes increases, the value of *Link Strength* increases. More nodes are selected for communication, increasing the connectivity and making all nodes participate in the network. The percentage increase in the value of *Link Strength* by changing the threshold from 0-1, 1-2 and 2-3 keymatches is about 30%, 7% and 7% respectively. In other words, as the threshold keys increase, the security increases.

7 CONCLUSIONS

In this paper, Unbalanced Distribution of Keys in Heterogeneous Networks is considered. Data routing in L1 and L2 nodes are determined. *Link Strength* is a factor of common keys that categorizes the paths as low, moderate and high secure path. The proposed Keymatches Constrained Secure Routing, *KCSR* algorithm discovers the secure and stable path which minimizes the effects of node compromise. Given the value of threshold security for a particular application, appropriate path may be chosen. Various degrees of security is seen in the same network. Switching to different paths during routing is permissible.

Mathematical analysis and Simulation results proved that *Link Strength vs threshold keys* has an exponential behavior. The new approach of routing provides enhanced resilience against node capture/compromise and allows us to select the required path. The performance of *KCSR* algorithm are justified by our extensive analysis and simulations.

REFERENCES

Chao, J. and Xiuli, R. (2008). A novel random key algorthim in wireless sensor networks. In *Proceedings of The International Conference on Advances in Mobile Computing and Multimedia*, pages 687–691.

Chris Karlof, N. S. and Wagner, D. (2004). Tinysec: A link layer security architecture for wireless sensor networks. In *Proceedings of ACM Conference on Simulators(SenSys)*, pages 162–175.

Debao Xiao, M. W. and Zhou, Y. (2006). Secure-spins: Secure sensor protocol for information *via* negotiation for wireless sensor networks. In *Proceedings of IEEE Conference on Industrial Electronics and Applications(ICIEA)*, pages 1–4.

Eschaenauer, L. and Gligor, V. (2002). A key management scheme for distributed sensor networks. In *Proceedings ACM Conference on Computer and Communication Security (CCS'02)*, pages 41–47.

F. L. Lewis, D. J. Cook, S. K. D. and Wiley, J. (2004). Wireless sensor networks. In *Smart Environment Technologies, Protocols and Applications, New York*, pages 1–18.

Haowen Chan, A. P. and Song, D. (2006). Random key predistribution schemes for sensor networks. In *Proceedings ACM/USENIX Fourth International Conference on Mobile Systems Applications and Services (Mobi Sys'06)*, pages 197–215.

Manel Boujelben, Omar Cheikhrouhon, M. A. and Youssef, H. (2009). Establishing pairwise keys in heterogeneous two-tiered wireless sensor networks. In *Proceedings Third International Conference on Sensor Technologies and Applications (Sensorcomm 2009)*, pages 442–448.

Perrig, A., S, R., W, V., C, D., and Tygar, J. D. (2002). Spins: Security protocols for sensor networks. In *Journal on Wireless Networks*, pages 521–534.

Rodrigo Roman, J. Z. and Lopez, J. (2005). On the security of wireless sensor networks. In *Proceedings ACM Conference on Network Security, Springer, LNCS, 3482*, pages 681–690.

Traynor, P., Kumar, R., Choi, H., Cao, G., Zhu, S., and LaPorta, T. (2007). Efficient hybrid security mechanisms for heterogeneous sensor networks. In *IEEE Transactions on Mobile Computing*, volume 6, pages 663–677.

Traynor, P., Kumar, R., Saad, H. B., Cao, G., and LaPorta, T. (2006). Liger: A hybrid key management scheme for heterogeneous sensor networks. In *Proceedings ACM/USENIX Fourth International Conference on Mobile Systems Applications and Services (Mobi Sys'06)*, pages 15–27.

Wang, Y. and Ramamurthy, B. (2008). A key management protocol for hybrid wireless sensor networks. In *Proceedings of The International Conference on Communication(ICC)*, pages 1625–1629.

Wenliang Du, Jing Deng, Y. S. H. and V, P. K. (2003). A pairwise key pre-distribution scheme for wireless sensor networks. In *Journal Computer and Communication Security*. ACM.

Yong Wang, B. R. and Xue, Y. (2008). A key management protocol for wireless sensor networks with multiple base station. In *Proceedings of IEEE International Conference(ICC 2008)*, pages 1625–1629.