Implementation Of Node Energy Based On Encryption Keying

Dr.S.Bhargavi
Electronics and Communication Engineering
S.J.C.I.T
Chikballapur, Karnataka, India

Ranjitha B.T
Electronics and Communication Engineering
S.J.C.I.T
Chikballapur, Karnataka, India

***Abstract*—This paper deals with Designing cost-efficient, secure network protocols for any Networks is a challenging problem because node in a network itself is resource-limited. Since the communication cost is the most dominant factor in any network, we introduce an energy-efficient Node Energy-Based Encryption and Keying (NEBEK) scheme that significantly reduces the number of transmissions needed for rekeying to avoid stale keys. NEBEK is a secure communication framework where sensed data is encoded using a scheme based on a permutation code generated via the RC4 encryption mechanism. The key to the RC4 encryption mechanism dynamically changes as a function of the residual energy of the node. Thus, a one-time dynamic key is employed for one packet only and different keys are used for the successive packets of the stream. The intermediate nodes along the path to the sink are able to verify the authenticity and integrity of the incoming packets using a predicted value of the key generated by the sender's virtual energy, thus requiring no need for specific reeking messages. NEBEK is able to efficiently detect and filter false data injected into the network by malicious outsiders. We have evaluated NEBEK's feasibility and performance analytically and through software simulations. Our results show that NEBEK, without incurring transmission overhead (increasing packet size or sending control messages for rekeying), is able to eliminate malicious data from the network in an energy efficient manner.**

Keywords- NEBEK; Network; protocol; communication; RC4 encryption; dynamic key; virtual energy; Statistical mode; Operational mode; Forwarding Node Packets.

I. Introduction

From a security point of view, it is important to provide authentic and accurate data to surrounding nodes and to the sink. Protocols should be such that they are resilient against false data injected into the network by malicious nodes. Else the consequences of propagating a false data in a network become costly, depleting the network resources and wasting responses. This becomes a challenging to the protocol builder in securing the network.

Here we focus on 2 keying mechanisms. Static and Dynamic keying. In static scheme keys are handling statistically. i.e. the network node will have fixed no of keys loaded. But dynamic after key revocation. Thus refreshed key doesn't become any stale key. Here we focus on minimizing the overhead associated with refreshing keys since the communication cost is the most dominant factor. This scheme performs keying function either periodically or on demand needed by the network. The major drawback of this keying mechanism is that it increases the communication overhead due to keys being refreshed in a network. Key refreshment may require for updating key.

In this project we develop an efficient and secure communication framework for network. Here we introduce NEBEK for network.

II. Literature Survey

A. Problem Statement

Sending confidential information from one node (source) to another node (destination) on a network could be a challenging task. Using the available resources and energy, the nodes exchange data of the received and sent packets and also ensure data integrity before it hits the sink.

The data exchanged could be manipulated or changed by the hacker on the network. So, the task would be to create a secure system that can ensure safety of the data using encryption methods (such as RC4) and still use the available energy and resources without much overhead.

B. Objective of the Paper

The objective of this paper is to discuss efficient and secure communication frameworks for Network applications by building upon the idea of sharing a dynamic cryptic credential.

Designing cost-efficient, secure network protocols for any Networks is a challenging problem because all the networks are resource-limited. Since the communication cost is the most dominant factor in a energy consumption, it is necessary to introduce an energy-efficient Node Energy-Based Encryption and Keying (NEBEK) scheme for LAN network that significantly reduces the number of transmissions needed for rekeying to avoid stale keys.

C. Existing System

An existing Dynamic Energy-based Encoding and Filtering (DEEF) framework is to detect the injection of false data into a sensor network. Dynamic Energy-based that each sensed event report be encoded using a simple encoding scheme based on a keyed hash.

The key to the hashing function dynamically changes as a function of the transient energy of the nodes, thus requiring no need for re-keying. Depending on the cost of transmission vs.

computational cost of encoding, it may be important to remove data as quickly as possible. Accordingly, DEEF can provide authentication at the edge of the network. Depending on the optimal configuration, as the report is forwarded, each node along the way verifies the correctness of the encoding probabilistically and drops those that are invalid.

Disadvantages

- Current schemes involve the usage of authentication keys and secret keys to disseminate the authentication keys; hence, it uses many keys and is complicated for resource-limited nodes.
- Current schemes are complicated for resource-constrained sensors as they transmit many keying messages in the network, which increases the energy consumption of WSNs that are already severely limited in the technical capabilities and resources (i.e., power, computational capacities, and memory) available to them.

D. Proposed System

NEBEK is a secure communication framework where the data is encoded using a scheme based on a permutation code generated via the RC4 encryption mechanism. The key to the RC4 encryption mechanism dynamically changes as a function of the residual energy of the network. Thus, a one-time dynamic key is employed for one packet only and different keys are used for the successive packets of the stream.

The intermediate nodes along the path to the sink are able to verify the authenticity and integrity of the incoming packets using a predicted value of the key generated by the sender's virtual energy, thus requiring no need for specific rekeying messages.

NEBEK's flexible architecture allows for adoption of stronger encryption mechanisms in lieu of encoding. And also show that our framework performs better than other comparable schemes in the literature with an overall 60-100 percent improvement in energy savings without the assumption of a reliable medium access control layer.

Advantages

- NEBEK's secure communication framework provides a technique to verify data in line and drop false packets from malicious nodes, thus maintaining the health of the wireless network.
- NEBEK dynamically updates keys without exchanging messages for key renewals and embeds integrity into packets as opposed to enlarging the packet by appending message authentication codes (MACs).
- The key to the encryption scheme (RC4) dynamically changes as a function of the residual virtual energy of the node, thus requiring no need for rekeying.
- The protocol is able to continue its operations under dire communication cases as it may be operating in a high-error-prone deployment area like under water.

III. System Description

A. Node Energy-Based Keying Module

NEBEK is a simple idea of designing the secure communication framework. It provides a technique to verify data in line and drop false packets from the malicious node, thus maintaining the security of network. Here data is encoded using RC4 encryption mechanism. RC4 mechanism dynamically changes as a function of residual energy of the network. The Node energy-based keying process involves the creation of dynamic keys. Here, it does not exchange extra messages to establish keys unlike other dynamic scheme methodologies. A node computes keys based on its residual energy of the network [5].

The rationale for using node energy as opposed to real battery levels as in our earlier work, DEEF [4], is that in reality battery levels may fluctuate and the differences in battery levels across nodes may spur synchronization problems, which can cause packet drops. These concerns have been addressed in NEBEK. After deployment, each nodes traverse several functional states. The states mainly include node-stay-alive, packet reception, transmission, encoding, and decoding. As each of these actions occur, the energy in a node is depleted. The current value of the node energy, Evc, in the node is used as the key to the key generation function, F. During the initial deployment, each node in a network will have the same energy level Eini, therefore, the initial key, K1, is a function of the initial virtual energy value and an initialization vector (IV).The IVs are pre distributed to the all the nodes. Subsequent keys, Kj, are a function of the current virtual energy, Evc, and the previous key Kj_1.

B. Operation mode of NEBEK

The NEBEK protocol provides three security services: Authentication, integrity and no repudiation. The fundamental idea behind providing these services is the watching mechanism. The watching Mechanism requires nodes to store one or more records (i.e. current energy level and Node-Id) to be able to compute the dynamic keys used by the source nodes, to decode packets, and to catch incorrect packets either due to communication Problems or potential attacks. However, there are costs (communication, computation, and storage) associated with providing these services. In reality, applications may have different security requirements. For instance, the security needed by a military application.

C. Operational mode

This is one of the operation mode in NEBEK. Here all nodes watch their neighbors, whenever a packet is received from a neighbor node, it is decoded and its authenticity and integrity are verified. Only valid or acceptable packets are forwarded toward the sink. In this mode, a short span of time exists at initial deployment so that no one can hack the network, because it takes time for an attacker to capture a node or get keys. During this period, information to initialize route, may be used by each node to decide which node to watch and a record is stored for each of its one-hop neighbor in its watch-list. To obtain a neighbor's initial energy value, a network-wise master key can be used to transmit this value during this period similar to the shared-key discovery phase of other dynamic key management schemes.

D. Statistical mode

In this operational mode, nodes in the network are configured to only watch some of the nodes in the network. Each node randomly picks node to monitor and stores the corresponding state before deployment. As a packet leaves the source node (originating node or forwarding node) it passes through node(s) that watch it based on probability. Thus, this method is a statistical filtering approach like SEF[7] and DEF[7]. If the current node is not watching the node that generated the packet, the packet is forwarded. If the node that generated the packet is being watched by the current node, the packet is decoded and the plaintext ID is compared with the decoded ID.

Similar to operational mode, if the watcher node wants to forward a packet and it cannot find the key successfully, it will try as many keys as the value of Key Search-threshold before actually classifying the packet as malicious. If the packet is authentic and the current hop is not the final destination then the original packet is forwarded, unless the current node is bridging the network. In the bridging case, the original packet is re encoded with the available bridge energy and forwarded. Since this node is bridging the network, both virtual and perceived energy values are decremented accordingly. If the packet is invalid or unacceptable, which is classified as such after exhausting all the virtual perceived energy values within the virtual Key Search Threshold window, the packet is discarded. This process continues until the packet reaches the sink.

E. Architecture model for NEBEK

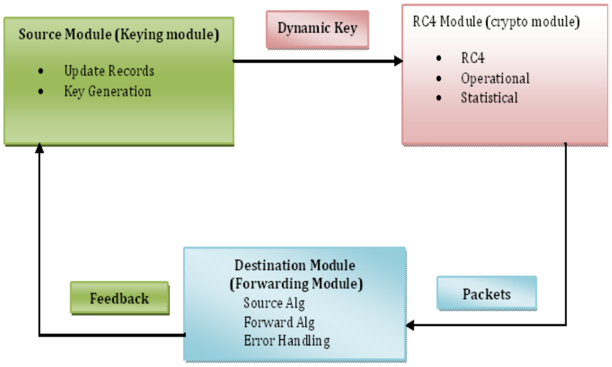

Figure1. Architecture Model for NEBEK

F. Source module (Keying Module)

The Node energy-based source module (keying module) of the NEBEK framework is one of the primary contribution of this project. It is essentially the method used for handling the keying process. It produces a dynamic key that is then fed into the RC4 module (crypto module).

In NEBEK, each node has a certain energy value when it is first deployed in the network. The reasons for using energy as opposed to real battery levels as in the DEEF is that in reality battery levels may fluctuate and the differences in battery levels across nodes may cause synchronization problems, which results in loosing packets.

After deployment, nodes travel across several functional states. The states mainly include node-stay-alive, packet reception, transmission, encoding, and decoding. As each of these actions occurs, the energy in a node is reduced. The current value of the energy, in the node is used as the key to the key generation function. During the initial deployment, each node will have the same energy level, therefore, the initial key, is a function of the initial energy value and an initialization vector. These are pre-distributed to the network. Subsequent keys are the result of the function of current energy and the previous key.

```
Algorithm: Compute Dynamic Key
ComputeDynamicKey(masterkey,packetsiz)
 begin
 j ← temp;
 if j → 1 then
        K← dynamickey(masterkey,packetsize)
 else
           K← dymamickey( kj-1, masterkey)
 end if
 return K
 end
```

Keying module ensures that each detected packet is associated with a new unique key generated based on the constantly changing value of the energy. After the dynamic key is generated, it is passed to the RC4 encryption module (crypto module), where the desired security services are implemented. The process of key generation is initiated when data is sensed, thus no explicit mechanism is needed to refresh or update keys. Because of the dynamic nature of the keys it makes difficult for attackers to prevent enough packets to break the encoding algorithm.

Each node computes and updates the constantly changing value of its energy after performing some actions. Each action on a node is associated with a certain predetermined cost. Since a node will be either forwarding some other nodes data or injecting its own data into the network, the set of actions and their associated energies for NEBEK includes packet reception, packet transmission, packet encoding, packet decoding energies, and the energy required to keep a node alive in the idle state.

G. RC4 Module (Crypto Module)

The RC4 (Crypto) module uses a simple encoding process, which is essentially the process of permutation of the bits in the packet according to the dynamically created permutation code generated via RC4. The encoding is a simple encryption mechanism adopted for NEBEK. However, NEBEK's flexible architecture allows for stronger encryption mechanisms in lieu of encoding.

In detail:

Due to the resource constraints of networks, traditional digital signatures or encryption mechanisms requiring expensive cryptography is not capable of doing what it is intended to do. The plan must be simple and effective. Thus a simple encoding operation is used [7]. The encoding operation is the process of permutation of the bits in the packet, according to the dynamically created permutation code via the RC4 encryption mechanism. The key to RC4 is created by the previous module (source or keying module). The purpose of the RC4 module is to provide simple confidentiality of the packet header and payload while ensuring the authenticity and integrity of sensed data without incurring transmission overhead of traditional schemes. However, since the key generation and handling process is done in another module, NEBEK's flexible architecture allows for adoption of stronger encryption mechanisms in lieu of encoding.

The packets in NEBEK consists of the ID (i-bits), type (t-bits) (assuming each node has a type identifier), and data (d-bits) fields. Each node sends these to its next hop. The nodes ID, type, and the sensed data are transmitted in a pseudorandom fashion according to the result of RC4.

The RC4 encryption algorithm takes the key and the packet fields (byte-by-byte) as inputs and produces the result as a permutation code shown in the Fig 2. The concatenation of each 8-bit output becomes the resultant permutation code. The key to the RC4 mechanism is taken from the keying module, which is responsible for generating the dynamic key according to the residual energy level.

The resultant permutation code is used in encoding the <ID|type|data> message. Then an additional copy of the ID is also transmitted along with the encoded message. The format of the final packet to be transmitted becomes Packet = [ID,{ID, type, data}k] where {x}k constitutes encoding x with key k. Thus instead of the traditional approach of sending the hash value (e.g., message digests and message authentication codes) along with the information to be sent, we use the result of the permutation code value. When the next node along the path to the sink receives the packet, it generates the local permutation code to decode the packet

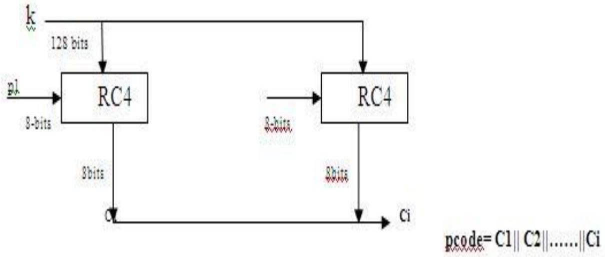

Figure2: RC4 encryption mechanism in NEBEK

Another significant step in the RC4 [8] or crypto module involves how the permutation code dictates the details of the encoding and decoding operations over the fields of the packet when generated by a source node or received by a forwarder node. Specifically the permutation code P can be mapped to a set of actions to be taken on the data stream combination.

The benefits of this simple encoding scheme are:

There is no hash code or message digest to transmit, the packet size does not grow, avoiding bandwidth overhead on an already resource-constrained network, thus increasing the network lifetime.

The technique is simple, thus ideal for devices with limited resources (e.g., PDAs).

The input to the RC4 encryption mechanism, the key, changes dynamically without sending control messages to rekey.

H. The Destination module (Forwarding Module)

The forwarding module handles the process of sending or receiving of encoded packets along the path to the sink.

The final module in the NEBEK communication architecture is the forwarding module. The forwarding module is responsible for the sending of packets (reports) initiated at the current node (source node) or received packets from other nodes (forwarding nodes) along the path to the sink.

The reports traverse the network through forwarding nodes and finally reach the terminating node, the sink. The operations of the forwarding module are explained in this section.

I. Source node Algorithm

When an event is detected by a source node, the next step is for the report to be secured. The source node uses the local virtual energy value and an Initial Vector (or previous key value if not the first transmission) to construct the next key. This dynamic key generation process is primarily handled by the source module. The source module fetches the current value of the virtual energy from the NEBEK module. The key is used as input into the RC4 algorithm inside the RC4 module to create a permutation code for encoding the <ID|type|data> message. The encoded message and the clear text ID of the originating node are transmitted to the next hop (forwarding node or sink) using the following format: [ID, {ID, type, data}Pc], where {x}Pc constitutes encoding x with permutation code Pc. The local virtual energy value is updated and stored for use with the transmission of the next report.

J. Forward node Algorithm

Once the forwarding node receives the packet it will first check its watch-list to determine if the packet came from a node it is watching. If the node is not being watched by the current node, the packet is forwarded without modification or authentication. Although this node performed actions on the packet (received and forwarded the packet), its local virtual perceived energy value is not updated. This is done to maintain synchronization with nodes watching it further up the route.

If the node is being watched by the current node, the forwarding node checks the associated current virtual energy record stored for the sending node and extracts the energy value to derive the key [6]. It then authenticates the message by decoding the message and comparing the plaintext node ID with the encoded node ID. If the packet is authentic, an updated energy value is stored in the record associated with the sending node. If the packet is not authentic it is discarded. The virtual energy value associated with the current sending node is only updated if this node has performed encoding on the packet.

IV. Software Implementation

.Net is used to implement this project. C# is a multi-paradigm programming language encompassing imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within the .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270). C# is one of the programming languages designed for the Common Language Infrastructure.

C# is intended to be a simple, modern, general-purpose, object-oriented programming language. Its development team is led by Anders Hejlsberg. The most recent version is C# 4.0, which was released on April 12, 2010.

Design goals

The ECMA standard lists these designgoals for C#

- C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.
- The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important.
- The language is intended for use in developing software components suitable for deployment in distributed environments.
- Source code portability is very important, as is programmer portability, especially for those programmers already familiar with C and C++.
- Support for internationalization is very important.
- C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.

Although C# applications are intended to be economical with regard to memory and processing power requirements, the language was not intended to compete directly on performance and size with C or assembly language.

A. *UML diagram*

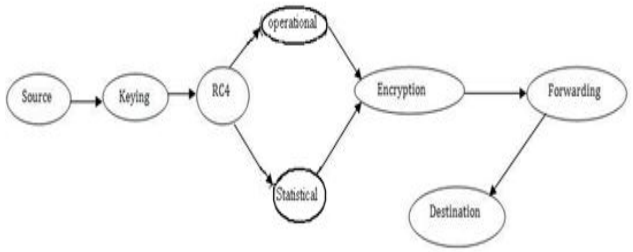

Figure 4: UML Diagram for NEBEK.

Figure 4 shows the UML diagram, various modules used in this project. It shows how the packets have been transferred from source to destination node that is the destination node. The packet will undergo various steps like keying, encoding, etc. Here in this project operational mode and statistical mode will take care of malicious packets.

B. *Data flow diagram*

The Figure 5 shows the flow of data in NEBEK.

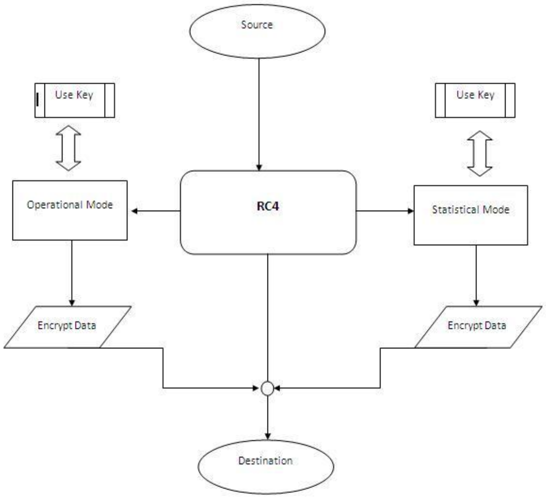

Figure5: Data Flow Diagram

The dynamic key will be generated in the source module and will be sent to RC4 module for encryption along with data to be sent. RC4 module encrypts each packet along with the unique dynamic key for each packet. The encrypted packet will be sent to operational or statistical mode to check the authenticity of the packet. If the packet are valid, they will be forwarded to the destination, where the packet will be decrypted. At the destination again the packet will be checked for authenticity and integrity of the packet.

V. Results And Discussion

Due to the broadcast nature of the networking medium, attackers may try to eavesdrop, intercept, or inject false messages. In this paper, we mainly consider the false injection and eavesdropping of messages from and outside malicious node; hence, insider attacks are outside the scope of this paper. This attacker is thought to have the correct frequency, protocol, and possibly a spoofed valid node ID.

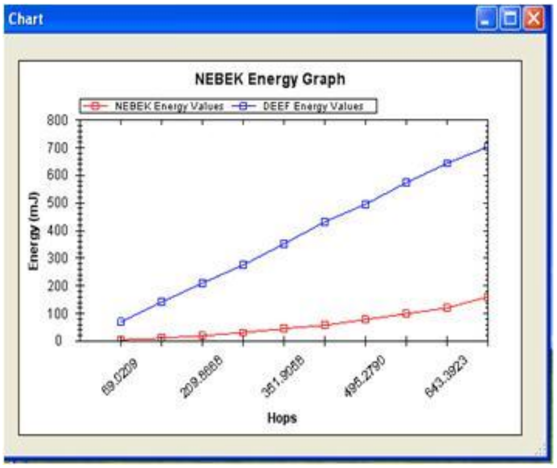

Figure6: Comparison of energy efficiency for NEBEK and DEEF

Filtering efficiency of statistical mode vs. operational mode

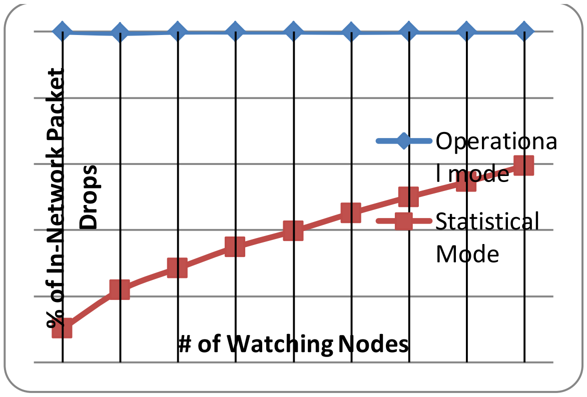

Figure7: Comparison of Modes of NEBEK

In Statistical and operational , in order for an attacker to be able to successfully inject a false packet, an attacker must forge the packet encoding (which is a result of dynamically created permutation code via RC4). Given that the complexity of the packet is 2l, [4]where l is the sum of the ID, TYPE, and DATA fields in the packet, the probability of an attacker correctly forging the packet [6] is:

P forg $= 1/2^{l}$ where l= packetsize

Accordingly, the probability of the hacker incorrectly forging the packet, and therefore, the packet being dropped

Ppdrop = 1- P forg

Since operational mode, authenticates at every hop, forged packets will always be dropped at the first hop with a probability of Ppdrop .

On the other hand, statistical mode, statistically drops packets along the route. Thus, the drop probability for statisticl mode, (Pdrop_II) is a function of the effectiveness of the watching nodes as well as the ability for a hacker to correctly guess the encoded packet structure. Accordingly, the probability of detecting and dropping a false packet at one hop when randomly choosing r records (nodes to watch) is:

P drop_II = $\frac{r}{N} * (1 - Pforg)$

Thus, the probability to detect and drop the packet when choosing r records after h hops is:

P pdrop_II = $1 - (1 - P\ drop_{II})^{h}$

Where h- Number of hops

r- Number of records.

Operational mode is always able to filter malicious packets from the network with its 100 percent filtering efficiency. This is mainly due to the fact that malicious packets are immediately taken out from the network at the next hop. However, the filtering efficiency of Statistical mode is closely related to the number of nodes (r) that each node watches.

VI. Conclusion

Communication is very costly for any network. Independent of the goal of saving energy, it may be very important to minimize the exchange of messages (e.g., military scenarios). To address these concerns, we presented a secure communication framework called Node Energy- Based Encryption and Keying. In comparison with other key management schemes, NEBEK has the following benefits: 1) it does not exchange control messages for key renewals and is therefore able to save more energy and is less chatty, 2) it uses one key per message so successive packets of the stream use different keys—making NEBEK more resilient to certain attacks (e.g., replay attacks, brute-force attacks, and masquerade attacks), and 3) it unbundled key generation from security services, providing a flexible modular architecture that allows for an easy adoption of different key-based encryption or hashing schemes. renewals and is therefore able to save more energy and is less.

References

[1] I.F.Akyildiz, W.Su, Y.Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey", Computer Networks vol. 38, no. 4, pp. 393-422, Mar. 2002.

[2] C.Vu,R.Beyah and Y. Li, "A Composite Event Detection in Wireless Sensor Networks," Proc. IEEE Int'l Performance, Computing and Comm. Conf. (IPCCC '07), Apr. 2007.

[3] G.J. Pottie and W.J. Kaiser, "Wireless Integrated Network Sensors," Comm. ACM, vol. 43, no. 5, pp. 51-58, 2000 Computerworld.

[4] H. Hou, C. Corbett, Y. Li, and R. Beyah, "Dynamic Energy-Based and Filtering in Sensor Networks", Proc. IEEE Military Comm. Conf. (MILCOM '07), Oct. 2007.

[5] Huy Hoang Ngo, Xianping Wu, Phu Dung Le, mpbell Wilson, and Balasubramaniam Srinivasan ,"Dynamic Key Cryptography and Applications," Monash University,900 Dandenong Road, Caul¯eld East,Victoria, 3145, Australia Feb. 9, 2009.

[6] Raheem A. Beyah, Yingshu Li, John A "Virtual Energy-Based Encryption and Keying for Wireless Sensor Networks".

[7] Ye, H. Luo, S. Lu, and L. Zhang, “Statistical En-Route Filtering of Injected False Data in Sensor Networks,”.

[8] Allam Mousa and Ahmad Hamad Electrical Engineering Department An-Najah University, “Evaluation of the RC4 Algorithm for Data Encryption “.

AUTHORS PROFILE

Dr.S.Bhargavi is presently working as a Professor in the department of Electronics and Communication engineering, SJCIT, Chikballapur, Karnataka, India. She is having 12 years of teaching experience. Her areas of interest are Robotics, Embedded Systems, Low Power VLSI, Wireless communication, ASIC and Cryptography.

Ranjitha B.T received Bachelor of Engineering degree in Computer Science from Visvesvaraya Technological University, Belgaum, Karnataka, India, in 2008. Currently she is pursuing M. Tech in Digital Communication and Network in

Visvesvaraya Technological University, Belgaum, Karnataka, India. She has 2 Years of Teaching experience. Her areas of interest are Computer Network, Cryptography and Wireless Communication.