International Journal of Research
ISSN: 2348-6848 Vol-3, Special Issue-1
National Conference on Advanced Computing Technologies
Held on 21st July 2015 at Hyderabad city organized by Global Research Academy, Hyderabad, Telangana, India.

Design & Development of Secure and Efficient Data Transmission Protocols for Cluster Based WSNs

K.Naveen Kumar[1]; Ms.Ch.Srilakshmi[2]& Prof.Dr.G.Manoj Someswar[3]

[1]M.Tech.(CSE) from Narasimha Reddy Engineering College, Affiliated to JNTUH, Hyderabad, Telangana, India

[2]M.Tech. (CSE), Associate Professor, Department of CSE, Narasimha Reddy Engineering College, Affiliated to JNTUH, Hyderabad, Telangana, India

[3]B.Tech., M.S.(USA), M.C.A., Ph.D., Principal & Professor, Department Of CSE, Anwar-ul-uloom College of Engineering & Technology, Affiliated to JNTUH, Vikarabad, Telangana, India

ABSTRACT:

Secure data transmission is a critical issue for wireless sensor networks (WSNs).Clustering is an effective and practical way to enhance the system performance of WSNs. In this research paper, we study a secure data transmission for cluster-based WSNs (CWSNs), where the clusters are formed dynamically and periodically. We propose two Secure and Efficient data Transmission (SET) protocols for CWSNs, called SET-IBS and SET-IBOOS, by using the Identity-Based digital Signature (IBS) scheme and the Identity-Based Online/Offline digital Signature (IBOOS) scheme, respectively. In SET-IBS, security relies on the hardness of the Diffie-Hellman problem in the pairing domain. SET-IBOOS further reduces the computational overhead for protocol security, which is crucial for WSNs, while its security relies on the hardness of the discrete logarithm problem. We show the feasibility of the SET-IBS and SET-IBOOS protocols with respect to the security requirements and security analysis against various attacks. The calculations and simulations are provided to illustrate the efficiency of the proposed protocols. The results show that the proposed protocols have better performance than the existing secure protocols for CWSNs, in terms of security overhead and energy consumption.

KEYWORDS: Wireless sensor network (WSN); Identity-Based Online/Offline digital Signature (IBOOS); Secure and Efficient data Transmission (SET); Condition-Based Maintenance (CBM)

INTRODUCTION

Distributed computing is a field of computer science that studies distributed systems. A distributed systemis a software system in which components located on networked computers communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal. There are many alternatives for the message passing mechanism, including RPC-like connectors and message queues. Three significant characteristics of distributed systems are: concurrency of components, lack of a global clock, and independent failure of components. An important goal and challenge of distributed systems is location transparency. Examples of distributed systems vary from SOA-based systems to massively multiplayer online games to peer-to-peer applications.

A computer program that runs in a distributed system is called a distributed program, and distributed programming is the process of writing such programs.[1]

Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers, which communicate with each other by message passing.

The word *distributed* in terms such as "distributed system", "distributed programming", and "distributed algorithm" originally referred to computer networks where individual computers were physically distributed within some geographical area. The terms are nowadays used in a much wider sense, even referring to autonomous processes that run on the same physical computer and interact with each other by message passing. While there is no single definition of a distributed system, the following defining properties are commonly used:[2]

- There are several autonomous computational entities, each of which has its own local memory.
- The entities communicate with each other by message passing.

In this article, the computational entities are called *computers* or *nodes.*[3]

A distributed system may have a common goal, such as solving a large computational problem.] Alternatively, each computer may have its own user with individual needs, and the purpose of the distributed system is to coordinate the use of shared resources or provide communication services to the users.Other typical properties of distributed systems include the following:

- The system has to tolerate failures in individual computers.
- The structure of the system (network topology, network latency, number of computers) is not known in advance, the system may consist of different kinds of computers and network links, and the system may change during the execution of a distributed program.
- Each computer has only a limited, incomplete view of the system. Each computer may know only one part of the input.

Distributed systems are groups of networked computers, which have the same goal for their work. The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:[4]

- In parallel computing, all processors may have access to a shared memory to exchange information between processors.
- In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.

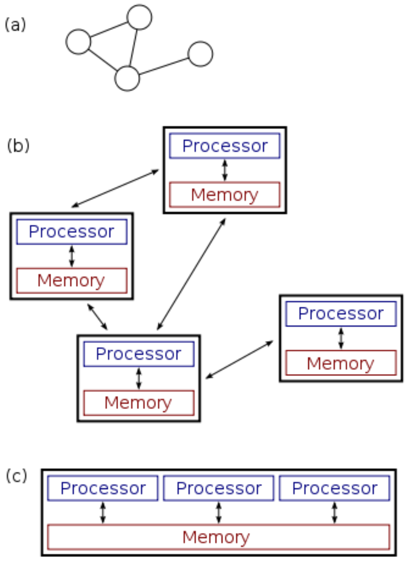

Figure 1: Difference Between Distributed and Parallel Systems

The figure on the right illustrates the difference between distributed and parallel systems. Figure (a) is a schematic view of a typical distributed system; as usual, the system is represented as a network topology in which each node is a computer and each line connecting the nodes is a communication link. Figure (b) shows the same distributed system in more detail: each computer has its own local memory, and information can be exchanged only by passing messages from one node to another by using the available communication links. Figure (c) shows a parallel system in which each processor has a direct access to a shared memory.[5]

The situation is further complicated by the traditional uses of the terms parallel and distributed *algorithm* that do not quite match the above definitions of parallel and distributed *systems*; see the section Theoretical foundations below for more detailed discussion. Nevertheless, as a rule of thumb, high-performance parallel computation in a shared-memory multiprocessor uses parallel algorithms while the coordination of a large-scale distributed system uses distributed algorithms.

WIRLESS SENSOR NETWORK:

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to *monitor* physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling *control* of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.[6]

Applications

Area monitoring

Area monitoring is a common application of WSNs. In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. A military example is the use of sensors detect enemy intrusion; a civilian example is the geo-fencing of gas or oil pipelines.

Environmental/Earth monitoring

The term Environmental Sensor Networks has evolved to cover many applications of WSNs to earth science research. This includes sensing volcanoes, oceans, glaciers, forests, etc. Some of the major areas are listed below.

Air quality monitoring

The degree of pollution in the air has to be measured frequently in order to safeguard people and the environment from any kind of damages due to air pollution. In dangerous surroundings, real time monitoring of harmful gases is an important process because the weather can change rapidly changing key quality parameters.[7]

- **Interior monitoring**

Observing the gas levels at vulnerable areas needs the usage of high-end, sophisticated equipment, capable to satisfy industrial regulations. Wireless internal monitoring solutions facilitate keep tabs on large areas as well as ensure the precise gas concentration degree.

- **Exterior monitoring**

External air quality monitoring needs the use of precise wireless sensors, rain & wind resistant solutions as well as energy reaping methods to assure extensive liberty to machine that will likely have tough access.

Air pollution monitoring

Wireless sensor networks have been deployed in several cities (Stockholm, London and Brisbane) to monitor the concentration of dangerous gases for citizens. These can take advantage of the ad hoc wireless links rather than wired installations, which also make them more mobile for testing readings in different areas. There are various architectures that can be used for such applications as well as different kinds of data analysis and data mining that can be conducted.

Forest fire detection

A network of Sensor Nodes can be installed in a forest to detect when a fire has started. The nodes can be equipped with sensors to measure temperature, humidity and gases which are produced by fire in the trees or vegetation. The early detection is crucial for a successful action of the firefighters; thanks to Wireless Sensor Networks, the fire brigade will be able to know when a fire is started and how it is spreading.[8]

Landslide detection

A landslide detection system makes use of a wireless sensor network to detect the slight movements of soil and changes in various parameters that may occur before or during a landslide. Through the data gathered it may be possible to know the occurrence of landslides long before it actually happens.

Water quality monitoring

Water quality monitoring involves analyzing water properties in dams, rivers, lakes & oceans, as well as underground water reserves. The use of many wireless distributed sensors enables the creation of a more accurate map of the water status, and allows the permanent deployment of monitoring stations in locations of difficult access, without the need of manual data retrieval.[9]

Natural disaster prevention

Wireless sensor networks can effectively act to prevent the consequences of natural disasters, like floods. Wireless nodes have successfully been deployed in rivers where changes of the water levels have to be monitored in real time.

Industrial monitoring

Machine health monitoring

Wireless sensor networks have been developed for machinery condition-based maintenance (CBM) as they offer significant cost savings and enable new functionality. In wired systems, the installation of enough sensors is often limited by the cost of wiring. Previously inaccessible locations, rotating machinery, hazardous or restricted areas, and mobile assets can now be reached with wireless sensors.

Data logging

Wireless sensor networks are also used for the collection of data for monitoring of environmental information; this can be as simple as the monitoring of the temperature in a fridge to the level of water in overflow tanks in nuclear power plants. The statistical information can then be used to show how systems have been working. The advantage of WSNs over conventional loggers is the "live" data feed that is possible.

Industrial sense and control applications

In recent research a vast number of wireless sensor network communication protocols have been developed. While previous research was primarily focused on power awareness, more recent research have begun to consider a wider range of aspects, such as wireless link reliability, real-time capabilities, or quality-of-service. These new aspects are considered as an enabler for future applications in industrial and related wireless sense and control applications, and partially replacing or enhancing conventional wire-based networks by WSN techniques.

Water/Waste water monitoring

Monitoring the quality and level of water includes many activities such as checking the quality of underground or surface water and ensuring a country’s water infrastructure for the benefit of both human and animal. The area of water quality monitoring utilizes wireless sensor networks and many manufacturers have launched fresh and advanced applications for the purpose.[14]

- **Observation of water quality**

The whole process includes examining water properties in rivers, dams, oceans, lakes and also in underground water resources. Wireless distributed sensors let users to make a precise map of the water condition as well as making permanent distribution of observing stations in areas of difficult access with no manual data recovery.

- **Water distribution network management**

Manufacturers of water distribution network sensors concentrate on observing the water management structures such as valve and pipes and also making remote access to water meter readings.[10]

- **Preventing natural disaster**

The consequences of natural perils like floods can be effectively prevented with wireless sensor networks. Wireless nodes are distributed in rivers so that changes of the water level can be effectively monitored.

Agriculture

Using wireless sensor networks within the agricultural industry is increasingly common; using a wireless network frees the farmer from the maintenance of wiring in a difficult environment. Gravity feed water systems can be monitored using pressure transmitters to monitor water tank levels, pumps can be controlled using wireless I/O devices and water use can be measured and wirelessly transmitted back to a central control center for billing. Irrigation automation enables more efficient water use and reduces waste.

- **Accurate agriculture**

Wireless sensor networks let users to make precise monitoring of the crop at the time of its growth. Hence, farmers can immediately know the state of the item at all its stages which will ease the decision process regarding the time of harvest.

- **Irrigation management**

When real time data is delivered, farmers are able to achieve intelligent irrigation. Data regarding the fields such as temperature level and soil moisture are delivered to farmers through wireless sensor networks. When each plant is joined with a personal irrigation system, farmers can pour the precise amount of water each plant needs and hence, reduce the cost and improve the quality of the end product. The networks can be employed to manage various actuators in the systems using no wired infrastructure.

- **Greenhouses**

Wireless sensor networks are also used to control the temperature and humidity levels inside commercial greenhouses. When the temperature and humidity drops below specific levels, the greenhouse manager must be notified via e-mail or cell phone text message, or host systems can trigger misting systems, open vents, turn on fans, or control a wide variety of system responses.

Recent research in wireless sensor networks in agriculture industry give emphasis on its use in greenhouses, particularly for big exploitations with definite crops. Such microclimatic ambiances need to preserve accurate weather status at all times. Moreover, using multiple distributed sensors will better control the above process, in open surface as well as in the soil.

Passive localization and tracking

The application of WSN to the passive localization and tracking of non-cooperative targets (i.e., people not wearing any tag) has been proposed by exploiting the pervasive and low-cost nature of such technology and the properties of the wireless links which are established in a meshed WSN infrastructure.

Smart home monitoring

Monitoring the activities performed in a smart home is achieved using wireless sensors embedded within everyday objects forming a WSN. State changes to objects based on human manipulation is captured by the wireless sensors network enabling activity-support services.

Characteristics

The main characteristics of a WSN include:

- Power consumption constrains for nodes using batteries or energy harvesting
- Ability to cope with node failures
- Mobility of nodes
- Communication failures
- Heterogeneity of nodes
- Scalability to large scale of deployment
- Ability to withstand harsh environmental conditions
- Ease of use

Sensor nodes can be imagined as small computers, extremely basic in terms of their interfaces and their components. They usually consist of a *processing unit* with limited computational power and limited memory, *sensors* or MEMS (including specific conditioning circuitry), a *communication device* (usually radio transceivers or alternatively optical), and a power source usually in the form of a battery. Other possible inclusions are energy harvesting modules, secondary ASICs, and possibly secondary communication interface (e.g. RS-232 or USB).

The base stations are one or more components of the WSN with much more computational, energy and communication resources. They act as a gateway between sensor nodes and the end user as they typically forward data from the WSN on to a server. Other special components in routing based networks are routers, designed to compute, calculate and distribute the routing tables.

Platforms

Standards and specifications

Several standards are currently either ratified or under development by organizations including WAVE2M for wireless sensor networks. There are a number of standardization bodies in the field of WSNs. The IEEE focuses on the physical and MAC layers; the Internet Engineering Task Force works on layers 3 and above. In addition to these, bodies such as the International Society of Automation provide vertical solutions, covering all protocol layers. Finally, there are also several non-standard, proprietary mechanisms and specifications.

Standards are used far less in WSNs than in other computing systems which makes most systems incapable of direct communication between different systems. However predominant standards commonly used in WSN communications include:

- ISA100.11a
- WirelessHART
- IEEE 1451
- ZigBee / 802.15.4
- ZigBee IP
- 6LoWPAN

Hardware

One major challenge in a WSN is to produce *low cost* and *tiny* sensor nodes. There are an increasing number of small companies producing WSN hardware and the commercial situation can be compared to home computing in the 1970s. Many of the nodes are still in the research and development stage, particularly their software. Also inherent to sensor network adoption is the use of very low power methods for data acquisition.

In many applications, a WSN communicates with over a Local Area Network or Wide Area Network through a gateway. The Gateway acts as a bridge between the WSN and the other network. This enables data to be stored and processed by device with more resources, for example in a remotely located Server_ (computing).

Software

Energy is the scarcest resource of WSN nodes, and it determines the lifetime of WSNs. WSNs are meant to be deployed in large numbers in various environments, including remote and hostile regions, where ad hoc communications are a key component. For this reason, algorithms and protocols need to address the following issues:

- Lifetime maximization
- Robustness and fault tolerance
- Self-configuration

Lifetime maximization: Energy/Power Consumption of the sensing device should be minimized and sensor nodes should be energy efficient since their limited energy resource determines their lifetime. To conserve power the node should shut off the radio power supply when not in use.

Some of the important topics in WSN (Wireless Sensor Networks) software research are:

- Operating systems
- Security
- Mobility

Operating systems

Operating systems for wireless sensor network nodes are typically less complex than general-purpose operating systems. They more strongly resemble embedded systems, for two reasons. First, wireless sensor networks are typically deployed with a particular application in mind, rather than as a general platform. Second, a need for low costs and low power leads most wireless sensor nodes to have low-power microcontrollers ensuring that mechanisms such as virtual memory are either unnecessary or too expensive to implement.

It is therefore possible to use embedded operating systems such as eCos or uC/OS for sensor networks. However, such operating systems are often designed with real-time properties.

TinyOS is perhaps the first operating system specifically designed for wireless sensor networks. TinyOS is based on an event-driven programming model instead of multithreading. TinyOS programs are composed of *event handlers* and *tasks* with run-to-completion semantics. When an external event occurs, such as an incoming data packet or a sensor reading, TinyOS signals the appropriate event handler to handle the event. Event handlers can post tasks that are scheduled by the TinyOS kernel some time later.[10]

LiteOS is a newly developed OS for wireless sensor networks, which provides UNIX-like abstraction and support for the C programming language.

Contiki is an OS which uses a simpler programming style in C while providing advances such as 6LoWPAN and Protothreads.

RIOT implements a microkernel architecture. It provides multithreading with standard API and allows for development in C/C++. RIOT supports common IoT protocols such as 6LoWPAN, IPv6, RPL, TCP, and UDP.

Online collaborative sensor data management platforms

Online collaborative sensor data management platforms are on-line database services that allow sensor owners to register and connect their devices to feed data into an online database for storage and also allow developers to connect to the database and build their own applications based on that data. Examples include Xively and the Wikisensing platform. Such platforms simplify online collaboration between users over diverse data sets ranging from energy and environment data to that collected from transport services. Other services include allowing developers to embed real-time graphs & widgets in websites; analyze and process historical data pulled from the data feeds; send real-time alerts from any data stream to control scripts, devices and environments.

The architecture of the Wiki sensing system is described in [21] describes the key components of such systems to include APIs and interfaces for online collaborators, a middleware containing the business logic needed for the sensor data management and processing and a storage model suitable for the efficient storage and retrieval of large volumes of data.

Simulation of WSNs

At present, agent-based modeling and simulation is the only paradigm which allows the simulation of complex behavior in the environments of wireless sensors (such as flocking). Agent-based simulation of wireless sensor and ad hoc networks is a relatively new paradigm. Agent-based modelling was originally based on social simulation.

Network simulators like OPNET, NetSim, NS2 and OMNeT can be used to simulate a wireless sensor network.

Other concepts

Distributed sensor network

If a centralized architecture is used in a sensor network and the central node fails, then the entire network will collapse, however the reliability of the sensor network can be increased by using a distributed control architecture. Distributed control is used in WSNs for the following reasons:

1. Sensor nodes are prone to failure,
2. For better collection of data
3. To provide nodes with backup in case of failure of the central node

There is also no centralized body to allocate the resources and they have to be self organized.

Data integration and Sensor Web

The data gathered from wireless sensor networks is usually saved in the form of numerical data in a central base station. Additionally, the Open Geospatial Consortium (OGC) is specifying standards for interoperability interfaces and metadata encodings that enable real time integration of heterogeneous sensor webs into the Internet, allowing any individual to monitor or control Wireless Sensor Networks through a Web Browser.

In-network processing

To reduce communication costs some algorithms remove or reduce nodes redundant sensor information and avoid forwarding data that is of no use. As nodes can inspect the data they forward they can measure averages or directionality for example of readings from other nodes. For example, in sensing and monitoring applications, it is generally the case that neighbouring sensor nodes monitoring an environmental feature typically register similar values. This kind of data redundancy due to the spatial correlation between sensor observations inspires the techniques for in-network data aggregation and mining.

LITERATURE SURVEY

Advances in wireless communication and electronics have enabled the development of low-cost, low power, multifunctional sensor nodes. These tiny sensor nodes, consisting of sensing, data processing, and communication components, make it possible to deploy Wireless Sensor Networks (WSNs), which represent a significant improvement over traditional wired sensor networks. WSNs can greatly simplify system design and operation, as the environment being monitored does not require the communication or energy infrastructure associated with wired networks [1]. WSNs are expected to be solutions to many applications, such as detecting and tracking the passage of troops and tanks on a battlefield, monitoring environmental pollutants, measuring traffic flows on roads, and tracking the location of personnel in a building. Many sensor networks have mission-critical tasks and thus require that security be considered [2, 3]. Improper use of information or using forged information may cause unwanted information leakage and provide inaccurate results. While some aspects of WSNs are similar to traditional wireless ad hoc networks, important distinctions exist which greatly affect how security is achieved.

Networking together hundreds or thousands of cheap micro sensor nodes allows users to accurately monitor a remote environment by intelligently combining the data from the individual nodes. These networks require robust wireless communication protocols that are energy efficient and provide low latency. We develop and analyze low-energy adaptive clustering hierarchy (LEACH), a protocol architecture for micro sensor networks that combines the ideas of energy-efficient cluster-based routing and media access together with application-specific data aggregation to achieve good performance in terms of system lifetime, latency, and application-perceived quality. LEACH includes a new, distributed cluster formation technique that enables self-organization of large numbers of nodes, algorithms for adapting clusters and rotating cluster head positions to evenly distribute the energy load among all the nodes, and techniques to enable distributed signal processing to save communication resources. Our results show that LEACH can improve system lifetime by an order of magnitude compared with general-purpose multihop approaches.[11]

Wireless sensor networks are a new class of ad hoc networks that will find increasing deployment in coming years, as they enable reliable monitoring and analysis of unfamiliar and untested environments. The advances in technology have made it possible to have extremely small, low powered sensor devices equipped with programmable computing, multiple parameter sensing, and wireless communication capability. Because of their inherent limitations, the protocols designed for such sensor networks must efficiently use both limited bandwidth and battery energy. We develop an M/G/1 model to analytically determine the delay incurred in handling various types of queries using our enhanced APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol) protocol. Our protocol uses an enhanced TDMA schedule to efficiently incorporate query handling, with a queuing mechanism for heavy loads. It also provides the additional flexibility of querying the network through any node in the network. To verify our analytical results, we have simulated a temperature sensing application with a Poisson arrival rate for queries on the network simulator ns-2. As the simulation and analytical results match perfectly well, this can be said to be the first step towards analytically determining the delay characteristics of a wireless sensor network.[12]

The main goal of this research is concerning clustering protocols to minimize the energy consumption of each node, and maximize the network lifetime of wireless sensor networks. However, most existing clustering protocols consume large amounts of energy, incurred by cluster formation overhead and fixed-level clustering, particularly when sensor nodes are densely deployed in wireless sensor networks. In this paper, we propose PEACH protocol, which is a power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks. By using overhearing characteristics of wireless communication, PEACH forms clusters without additional overhead and supports adaptive multi-level clustering. In addition, PEACH can be used for both location-unaware and location-aware wireless sensor networks. The simulation results demonstrate that PEACH significantly minimizes energy consumption of each node and extends the network lifetime, compared with existing clustering protocols. The performance of PEACH is less affected by the distribution of sensor nodes than other clustering protocols.[13]

Recently, Wireless Sensor Network (WSN) has been broadly studied in ubiquitous computing environments. In WSN, it is important to reduce communication overhead by using an energy-efficient routing protocol because the resources of the sensor node are limited. Although there exist some cluster-based routing protocols, they have some problems. First, the random selection of a cluster head incurs a node concentration problem. Secondly, they have a low reliability for data communication due to the less consideration of node communication range. Finally, data communication overhead is greatly increased while constructing clusters. To solve these problems, we, in this paper, propose a new cluster-based routing protocol using message success rate. To resolve the node concentration problem, we design a new cluster head selection algorithm based on node connectivity and devise cluster maintenance algorithms. Moreover, to guarantee data communication reliability, we use message success rate, which is one of popular measures for data communication reliability, in order to select a routing path. Finally, to reduce data communication overhead, we use only the information of neighboring nodes during both cluster construction and cluster head selection phases. Through our performance analysis, we show that our protocol outperforms existing schemes in terms of communication reliability and energy efficiency.[14]

Clustering protocols are often used in sensor networks. In many deployment scenarios, security is a key concern. In this paper we provide a secure solution to a commonly used clustering protocol, the LEACH protocol. We show that our protocol, the GS-LEACH protocol is more energy efficient than any of the secure flavors of LEACH. The GS-LEACH (grid-based secure LEACH) protocol uses pre deployment key distribution using prior knowledge of the deployment area. We also provide a detailed security analysis of our protocol and show that it is more secure than the secure versions of LEACH. Finally with the results of our simulation experiments we show that our protocol is very energy efficient and provides a longer network lifetime compared to the other flavors of LEACH.[15]

SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM DESIGN

SYSTEM ARCHITECTURE:

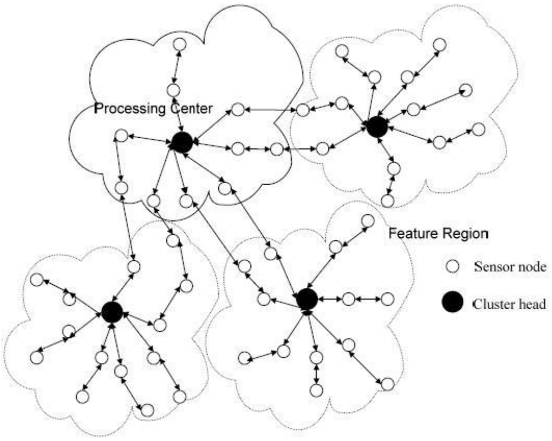

Figure 2: System Architecture

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

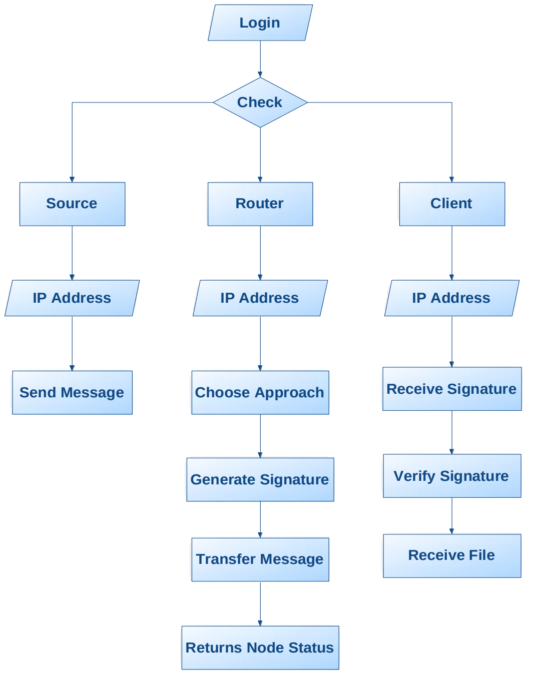

Figure 3: Data Flow Diagram

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

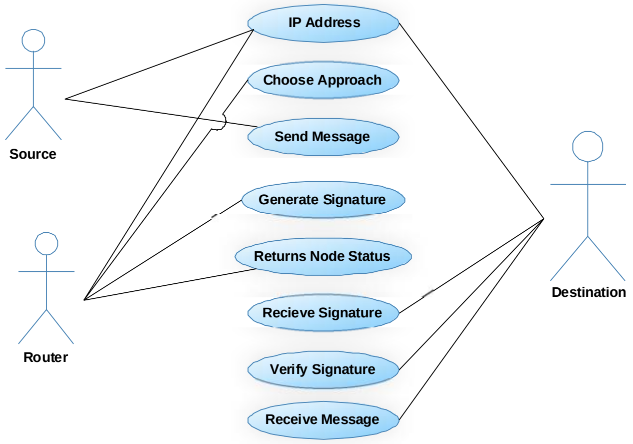

Figure 4: Use Case Diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

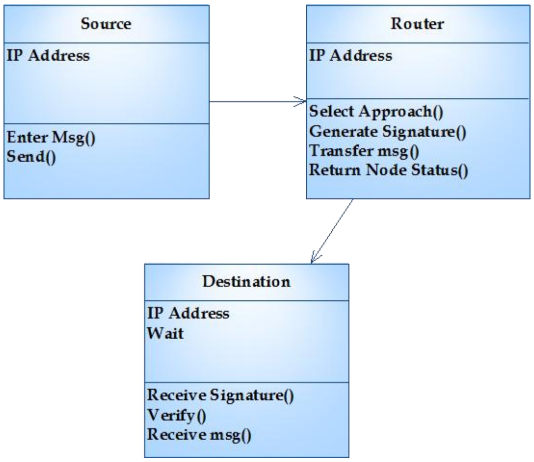

Figure 5: Class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

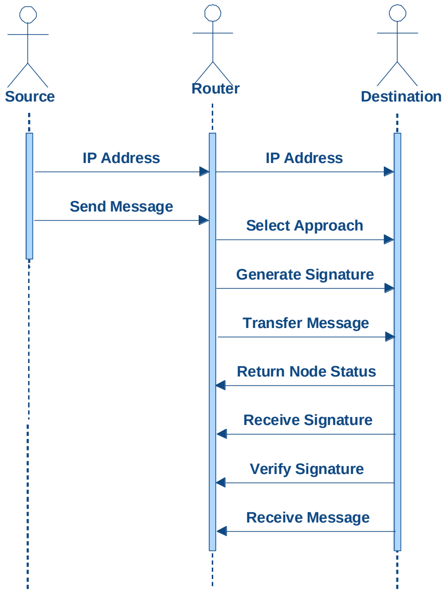

Figure 6: Sequence Diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

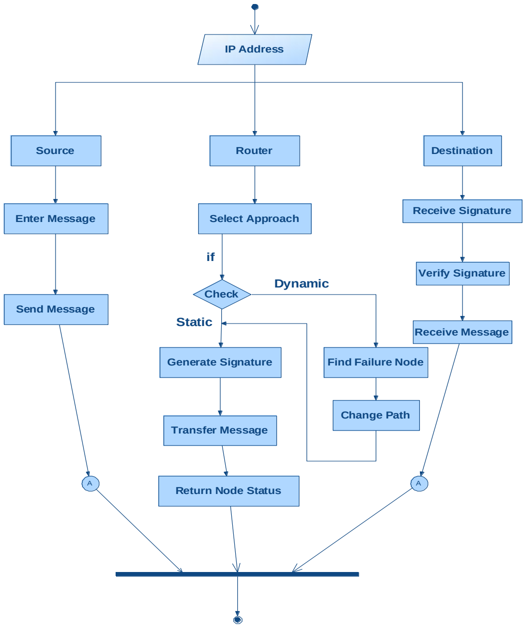

Figure 7:Activity Diagram

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2.It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the userwill not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

SYSTEM ANALYSIS

EXISTING SYSTEM:

In this Existing System of wireless sensor network comprised of spatially distributed devices using wireless sensor nodes to monitor physical or environmental conditions, such as sound, temperature, and motion. The individual nodes are capable of sensing their environments, processing the information data locally, and sending data to one or more collection points in a WSN.

Efficient data transmission is one of the most important issues for WSNs. Meanwhile, many WSNs are deployed in harsh, neglected and often adversarial physical environments for certain applications, such as military domains and sensing tasks with trustless surroundings.

DISADVANTAGES OF EXISTING SYSTEM:

- The clusters are formed dynamically and periodically.
- Existing solutions are provided for distributed WSNs, but not for CWSNs.
- It reduces the possibility of a node joining with a CH.
- Problem occurs when a node does not share a pairwise key with others in its preloaded key ring.

PROPOSED SYSTEM:

In this Proposed System, Secure and efficient data transmission is thus especially necessary and is demanded in many such practical WSNs. So, we propose two Secure and Efficient data Transmission (SET) protocols for CWSNs, called SET-IBS and SET-IBOOS, by using the Identity-Based digital Signature (IBS) scheme and the Identity-Based Online/Offline digital Signature (IBOOS) scheme, respectively.

It has been proposed in order to reduce the computation and storage costs to authenticate the encrypted sensed data, by applying digital signatures to message packets, which are efficient in communication and applying the key management for security.

In the proposed protocols pairing parameters are distributed and preloaded in all sensor nodes by the BS initially.

ADVANTAGES OF PROPOSED SYSTEM:

- Overcomes the key escrow problem described in ID-based cryptosystems.
- Efficient in communication and saves energy.
- Solve the orphan node problem in the secure data transmission with a symmetric key management.
- More feasibile.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

Verify that the entries are of the correct format

- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

IMPLEMENTATION

MODULES:

- Network Architecture
- IBS scheme
- IBOOS Scheme
- Key Management

MODULES DESCRIPTION:

Network Architecture

Consider a CWSN consisting of a fixed BS and a large number of wireless sensor nodes, which are homogeneous in functionalities and capabilities. We assume that the BS is always reliable, i.e., the BS is a trusted authority (TA). Meanwhile, the sensor nodes may be compromised by attackers, and the data transmission may be interrupted from attacks on wireless channel. In a CWSN, sensor nodes are grouped into clusters, and each cluster has a CH sensor node, which is elected autonomously. Leaf (non-CH) sensor nodes join a cluster depending on the receiving signal strength and transmit the sensed data to the BS via CHs to save energy. The CHs perform data fusion, and transmit data to the BS directly with comparatively high energy. In addition, we assume that all sensor nodes and the BS are time synchronized with symmetric radio channels, nodes are distributed randomly, and their energy is constrained.

IBS scheme

An IBS scheme implemented for CWSNs consists of the following operations, specifically, setup at the BS, key extraction and signature signing at the data sending nodes, and verification at the data receiving nodes

IBOOS Scheme

An IBOOS scheme implemented for CWSNs consists of following four operations, specifically, setup at the BS, key extraction and offline signing at the CHs, online signing at the data sending nodes, and verification at the receiving nodes

Key Management

Assume that a leaf sensor node j transmits a message M to its CH i, and encrypts the data using the encryption key k from the additively homomorphic encryption scheme. We denote the ciphertext of the encrypted message as C. We adapt the algorithms of the IBS scheme from to CWSNs practically and provide the full algorithm in the signature verification, where security is based on the DHP in the multiplicative group. The IBS scheme in the proposed SET-IBS consists of following three operations: extraction, signing, and verification.

RESULTS &CONCLUSION

In this paper, we first reviewed the data transmission issues and the security issues in CWSNs. The deficiency of the symmetric key management for secure data transmission has been discussed. We then presented two secure and efficient data transmission protocols, respectively, for CWSNs, SET-IBS, and SET-IBOOS. In the evaluation section, we provided feasibility of the proposed SET-IBS and SET-IBOOS with respect to the security requirements and analysis against routing attacks. SET-IBS and SETIBOOS are efficient in communication and applying the IDbased cryptosystem, which achieves security requirements in CWSNs, as well as solved the orphan node problem in the secure transmission protocols with the symmetric key management. Lastly, the comparison in the calculation and simulation results show that the proposed SET-IBS and SET-IBOOS protocols have better performance than existing secure protocols for CWSNs. With respect to both computation and communication costs, we pointed out the merits that using SET-IBOOS with less auxiliary security overhead is preferred for secure data transmission in CWSNs.

REFERENCES

[1] T. Hara, V.I. Zadorozhny, and E. Buchmann, Wireless Sensor Network Technologies for the Information Explosion Era, Studies in Computational Intelligence, vol. 278. Springer-Verlag, 2010.

[2] Y. Wang, G. Attebury, and B. Ramamurthy, “A Survey of Security Issues in Wireless Sensor Networks,” IEEE Comm. Surveys & Tutorials, vol. 8, no. 2, pp. 2-23, Second Quarter 2006.

[3] A.A. Abbasi and M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks," Computer Comm., vol. 30, nos. 14/ 15, pp. 2826-2841, 2007.

[4] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Trans. Wireless Comm., vol. 1, no. 4, pp. 660-670, Oct. 2002.

[5] A. Manjeshwar, Q.-A. Zeng, and D.P. Agrawal, "An Analytical Model for Information Retrieval in Wireless Sensor Networks Using Enhanced APTEEN Protocol," IEEE Trans. Parallel & Distributed Systems, vol. 13, no. 12, pp. 1290-1302, Dec. 2002.

[6] S. Yi et al., "PEACH: Power-Efficient and Adaptive Clustering Hierarchy Protocol for Wireless Sensor Networks," Computer Comm., vol. 30, nos. 14/15, pp. 2842-2852, 2007.

[7] K. Pradeepa, W.R. Anne, and S. Duraisamy, "Design and Implementation Issues of Clustering in Wireless Sensor Networks," Int'l J. Computer Applications, vol. 47, no. 11, pp. 23-28, 2012.

[8] L.B. Oliveira et al., "SecLEACH-On the Security of Clustered Sensor Networks," Signal Processing, vol. 87, pp. 2882-2895, 2007.

[9] P. Banerjee, D. Jacobson, and S. Lahiri, "Security and Performance Analysis of a Secure Clustering Protocol for Sensor Networks," Proc. IEEE Sixth Int'l Symp. Network Computing and Applications (NCA), pp. 145-152, 2007.

[10] K. Zhang, C. Wang, and C. Wang, "A Secure Routing Protocol for Cluster-Based Wireless Sensor Networks Using Group Key Management," Proc. Fourth Int'l Conf. Wireless Comm., Networking and Mobile Computing (WiCOM), pp. 1-5, 2008.

[11] S. Sharma and S.K. Jena, "A Survey on Secure Hierarchical Routing Protocols in Wireless Sensor Networks," Proc. Int'l Conf. Comm., Computing & Security (ICCCS), pp. 146-151, 2011.

[12] G. Gaubatz et al., "State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks," Proc. IEEE Third Int'l Conf. Pervasive Computing and Comm. Workshops (PerCom), pp. 146-150, 2005.

[13] W. Diffie and M. Hellman, "New Directions in Cryptography," IEEE Trans. Information Theory, vol. IT-22, no. 6, pp. 644-654, Nov. 1976.

[14] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," Proc. Advances in Cryptology (CRYPTO), pp. 47-53, 1985.

[15] D.W. Carman, "New Directions in Sensor Network Key Management," Int'l J. Distributed Sensor Networks, vol. 1, pp. 3-15, 2005.