International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 2, Issue 2, February 2015

WIRELESS SENSOR NETWORKS PROTOCOL ANALYSIS AND PERFORMANCE SIMULATION

ANTO JESSY MARY.A[1], ARULRAJ[2]

Abstract **Wireless Sensor Networks (WSN) are wireless networks composed of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. Each sensor node in a sensor network is typically equipped with a radio transceiver or other wireless communication device, a small microcontroller, and an energy source, usually a battery. The security analysis and performance simulation show that ROSS not only achieves the predefined security goals, but also allows a tradeoff between security and performance cost. In three new mechanisms in the framework of random key redistributions were proposed to address the bootstrapping problem, namely the q-composite scheme, the multipath reinforcement scheme, and the random pair wise scheme. Each of these three schemes represents a different tradeoff in the design space of random key protocols.**

Index Terms Sense temperature, pressure, sound, light, vibration.

I. INTRODUCTION

Although the development of WSNs were originally motivated by military applications such as battlefield surveillance, however, due to the deployment flexibility and maintenance simplicity, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control. As the applications gain more ground, security issues have also become a hot research topic. In a Resource Oriented Security Solution (ROSS) was introduced to protect the network connectivity of heterogeneous

Clustered sensor networks. The security analysis and performance simulation show that ROSS not only achieves the predefined security goals, but also allows a tradeoff between security and performance cost. In three new mechanisms in the framework of random key pre-distribution were proposed to address the bootstrapping problem, namely the q-composite scheme, the multipath reinforcement scheme, and the random pair wise scheme. Each of these three schemes represents a different tradeoff in the design space of random key protocols.[2]

1.2 KEY MANAGEMENT SCHEME

A key management scheme for distributed sensor networks. The scheme is designed to satisfy both operational and security requirements. It relies on probabilistic key sharing among the nodes of a random graph and uses simple protocols for shared-key discovery and path-key establishment, and for key revocation, re-keying, and incremental addition of nodes. The localized broadcast authentication in large sensor networks to take full advantage of roles of network nodes and local information in a sensor network, and to provide an alternative solutions regarding tradeoff between verification delay and broadcast overhead for satisfying applications with different requirements.To construct the shared session key in wireless sensor network. This protocol has great scalability in simulation because the time needed to finish key negotiation does not depend on the number the sensor nodes. [11] It can also save power by reducing the number of transmissions. The security attributes of wireless ad hoc networks were discussed. A WSN has some unique characteristics, such as limited computing power, mobility of nodes, large scale of deployment, unattended operation, limited communication bandwidth, and limited storage resources. These characteristics plus high risk of physical attacks to unattended sensors pose challenges to security in WSN. Moreover, in some deployment scenarios sensor nodes need to operate under an adversarial condition. Security solutions for such applications depend on the existence of strong and efficient key distribution mechanisms

1.3 DYNAMIC KEY MANAGEMENT

A dynamic key management protocol is proposed to satisfactorily resolve the above two issues. The protocol assumes that the wireless sensor system has already been equipped with effective security detection mechanisms, which can decide if a sensor node is compromised or has used up its energy.

2.0RESOURCEORIENTED SECURITY SOLUTION FOR HETEROGENEOUS CLUSTERED SENSOR NETWORKS

We measure the suspiciousness of a large node with the number of alarms against it. Whenever a neighbor node determines that a particular large node is compromised, it reports an alarm to the BS. Since malicious nodes may report many alarms against benign large nodes, we limit the number of alarms each large node can report to mitigate this effect. The BS maintains an alarm counter and a report counter for each large node. The alarm counter records the suspiciousness of this large node, while the report counter records the number of alarms this node reported and accepted by the BS. Most existing security schemes of sensor networks are not suitable for HCSNs because of the unique properties of HCSNs.

In this paper, we propose ROSS; a resource oriented network-layer security solution which is customized for HCSNs. ROSS contains proactive cryptography mechanisms and reactive mechanisms to provide a multiple defense solution for the network layer. Both security analysis and performance cost evaluation have confirmed that the effectiveness and efficiency of ROSS meet the predefined standard in protecting the network layer of HCSNs. However, ROSS is designed for static sensor networks where nodes cannot move.

2.1 PROPOSED SYSTEM

Recent advancement in wireless communication and microelectronics has enabled the design and development of wireless sensor networks with low cost, low energy consumption and high utilization. Many cluster-based wireless sensor network routing protocols have been proposed. However, most of them take little consideration on communication protection, which is important to ensure the network security. In this paper, a lightweight key management approach is presented. Its analysis shows that this approach is an effective solution to the key management of hierarchical clustered wireless sensor networks.

3.0 SYSTEM DESIGN

System design contains Logical Design & Physical Designing; logical designing describes the structure & characteristics or features, like output, input, files, database & procedures. The physical design, which follows the logical design, actual software & a working system. There will be constraints like Hardware, Software, Cost, and Time & Interfaces.

3.1 SOFTWARE 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.

4.0 MAINTENANCE

Correction-Correct defects in the software: defects relating to incorrect requirements, or incorrectly specifications; defects from any of the construction phases - `bugs'.

Adaption -Adapt to changes in the original software and hardware platform. E.g. simpler: MS-DOS to Windows. Complex: stand-alone to client-server.

Enhancement-Customer identifies additional requirements.

Prevention-After many sets of changes, the software `deteriorates', or otherwise becomes difficult to maintain. See **Reengineering**, **Legacy Systems**. Problems:

*Programmers do not like doing maintenance.

*Maintenance needs a project of its own. It is very uncommon to include maintenance in a (development) contract.

*Cost estimation. Normal software cost estimation is difficult enough; most *Estimation models do not include maintenance.

*Design deficiencies make system impossible to extend.

*Deterioration of legacy systems.

5.0 UNIT TESTING

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. [5]The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.1 **INTEGRATION TESTING**

Testing in which software components, hardware components, or both together are combined and tested to evaluate interactions between them. Integration testing takes as its input modules that have been checked out by unit testing, groups them in larger aggregates, applies tests defined in an Integration test

plan to those aggregates, and delivers as its output the integrated system ready for system testing.

5.2REGRESSION TESTING-Regression testing is the process of testing changes to computer programs to make sure that the older programming still works with the new changes. Regression testing is a normal part of the program development process. Test department coders develop code test scenarios and exercises that will test new units of code after they have been written. [3]Before a new version of a software product is released, the old test cases are run against the new version to make sure that all the old capabilities still work. The reason they might not work because changing or adding new code to a program can easily introduce errors into code that is not intended to be changed. It is a quality control measure to ensure that the newly modified code still complies with its specified requirements and that unmodified code has not been affected by the maintenance activity.

5.3ACCEPTANCE TESTING-Acceptance testing is formal testing conducted to determine whether a system satisfies its acceptance criteria and thus whether the customer should accept the system.

The main types of software testing are:

- Component.
- Interface.
- System.
- Acceptance.
- Release.

Acceptance Testing checks the system against the "Requirements". It is similar to systems testing in that the whole system is checked but the important difference is the change in focus:Systems testing checks that the system that was specified has been delivered. Acceptance Testing checks that the system delivers what was requested. The customer and not the developer should always do acceptance testing. The customer knows what is required from the system to achieve value in the business and is the only person qualified to make that judgment. The User Acceptance Test Plan will vary from system to system but, in general, the testing should be planned in order to provide a realistic and adequate exposure of the system to all reasonably expected events. The testing can be based upon the User Requirements Specification to which the system should conform.

5.4 FUNCTIONAL TESTING – This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

5.5 SYSTEM TESTING – Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

5.6 REGRESSION TESTING – Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

5.7 PERFORMANCE TESTING – Term often used interchangeably with 'stresses and 'load' testing. To check whether system meets performance requirements. Used different performance and load tools to do this.

5.8 INSTALL/UNINSTALL TESTING - Tested for full, partial, or upgrade install/uninstall processes on different operating systems under different hardware, software environment.

5.9 RECOVERY TESTING – Testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

5.10 SECURITY TESTING – Can system be penetrated by any hacking way? Testing how well the system protects against unauthorized internal or external access. Checked if system database is safe from external attacks.

6.0 Source Code

```
//package Login;
import java.sql.*;
import javax.swing.*;
import java.io.*;
import java.net.*;
import java.io.*;
import java.util.*;
public class Admin
{
JLabel l1,l2,l3,l4,l5,l6,l7;
JList li1,li2;
JList lis1,lis2;
JTextField tf1;
JButton b1,b2,b3,b4;
Container panel1;
JComboBox cb1;
String RH;
JScrollPane sb1,sb2,sb3,sb4;
DataBase_Connection DBC=new
DataBase_Connection();
JFrame f=new JFrame();
Vector cli=new Vector();
Vector conn=new Vector();//package Login;
import java.awt.Dimension;
import java.awt.Point;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import java.net.*;
import com.birosoft.liquid.LiquidLookAndFeel;
public class Registration
```

```
{
JLabel l1,l2,l3,l4,l5;
JTextField t1,t2,t3;
JButton b1,b2;
Container panel1;
Socket socket;
JFrame f=new JFrame();
Clientipaddress ca=new Clientipaddress();
sysname a = new sysname();
String ip= a.orgin();
String strSRep;
String PName,IPAdd,sPort,strReq,strRe,strRep;
public static ServerConnection sc =new
ServerConnection();
public Registration()
{ initialization1();
try {
}
catch (Exception e1)
{
System.out.println(e1);
System.out.println("end");
}
}
public void initialization1()
{ l1=new JLabel("Node Registration");
 l2=new JLabel("Name");
 l3=new JLabel("IPAddress");
 l4=new JLabel("PortNumber");
 l5=new JLabel(new ImageIcon("loginimg.jpg"));
 t1=new JTextField();
 t2=new JTextField(ip);
 t3=new JTextField();
 b1=new JButton("Register");
// b2=new JButton("Complete");
// l5.setIcon(new ImageIcon("bg.bmp"));
 f=new
JFrame();panel1=(JPanel)f.getContentPane();
//panel1.setBackground(new Color(102,138,150));

 t2.setEnabled(false);
f.setLayout(null);
addComponent(panel1, l5, 0,0,400,200) /*
 b2.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e)
{
b2_actionPerformed(e);
}
}); */
b1.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e)
{
b1_actionPerformed(e);
}
});
f.setTitle("Node Registration");f.setLocation(new
Point(50, 100));
f.setSize(new Dimension(400,500));
f.setDefaultCloseOperation(WindowConstants.EXIT_O
N_CLOSE);
f.setVisible(true);
}
public static void main(String args[])
{
try{
//UIManager.setLookAndFeel("com.birosoft.liquid.Liq
uidLookAndFeel");
UIManager.setLookAndFeel(
//UIManager.getCrossPlatformLookAndFeelClassNam
e());
UIManager.getSystemLookAndFeelClassName());
}catch(Exception e){
System.out.println("expc"+e.toString());
 }
Registration reg=new Registration();
}
}
Vector cli2=new Vector();
Vector tl=new Vector();
Vector ts=new Vector();
private void b1_actionPerformed(ActionEvent e)
{
try
{
Vector trss=new Vector();
String strSp[]=RH.split("@");
DataBase_Connection DBC=new
DataBase_Connection();
System.out.println(strSp[0]+""+strSp[1]+""+strSp[2]);
 ts=DBC.RHs(strSp[0],strSp[1],strSp[2]);
 lis2.setListData(ts);
 trss=DBC.HH1();
//tl.add(strNode1+"@"+strNNode1+"@"+strNNode2);
lis1.setListData(trss) }
catch (Exception ef)
{
}
```

7.0 CONCLUSION

As the applications of wireless sensor networks gain more ground, security issues have also become a hot research topic. This paper discussed the clustered WSN key management protocols and proposed a new protocol which is suitable for the key management of dynamic clustered networks, based on their operation mechanisms. The proposed protocol addresses the network security issues with cluster head update. It is distinguished with low power consumption, less computation workload and enhanced security.

Besides, the protocol uses a symmetric key system, and consists of the sub-protocols that define how keys are distributed, added, revoked, and updated during the life time of the sensor network. The protocol assumes that each sensor node is able to get its location information, which is currently a major restriction to its application.

8.0 REFERENCES

[1] X. Cao and G. Chen, (2007) "ROSS: Resource Oriented Security Solution for Heterogeneous Clustered Sensor Networks," Int. J. of Intelligent Control and Systems, 12(4):317- 324,.

[2] H. W. Chan, A. Perrig, and D. Song, (2003)"Random Key Predistribution Schemes for Sensor Networks," in: Proceedings of IEEE Symp. on Security and Privacy, 197-215, Berkeley, CA.

[3] M. Eltoweissy, M. Heydari, L. Morales, and H. Sudborough, , (2004) "Combinatorial Optimization for Key Management in Secure Multicast Environments," Journal of Network and System Management, 12(1):33-50.

[4] M. Eltoweissy, M. Younis, and K. Ghumman, (2004) "Lightweight Key Management for Wireless Sensor Networks," IEEE International Conference on Performance Computing and Communications, 813-818.

[5] L. Eschenauer, Virgil.D Gligor, (2002) "A Key Management Scheme for Distributed Sensor Networks," in: Proceedings of the 9th ACM Conference on Computer and Communication Security, Washington DC.

[6] Q. Gu, and J. Drissi,(2007) Localized Broadcast Authentication in Large Sensor Networks, Int. J. of Intelligent Control and Systems, 12(4): 341- 350.

[7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan,(2000) "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in: Proceedings of the 33rd Annual Hawaii Int'l Conf. on System Sciences. Maui: IEEE Computer Society, 3005−3014,.

[8] G. Jolly, M. Kuscu, P. Kokate, and M. Younus, (2003) "A Low-Energy Key Management Protocol for Wireless Sensor Networks," in: Proceedings of the 8th IEEE Symposium on Computer and Communications (ISCC), Antalya, 335-340.

**ANTO JESSY MARY.A ***is working as lecturer in FMC college-Madikeri under Mangalore University in dept. of computer science .i participated and presented 4 national and 2 international papers in relevant area and computational fluid dynamics mechanical software.

I.Arulraj is a research scholar in the Faculty of Mechanical Engineering at Sathyabama University, Chennai, India. I am working as a Assistant Professor in the department of Mechanical Engineering, St.Mother Theresa Engineering College-Vagaikulam-Thoothukudi Dt.Tamilnadu. I have completed my Bachelor degree in Mechanical Engineering in the year 1999 and Master degree in Thermal engg. in the year 2006.I have more than 10 years of experience in the teaching field. I have more than 4 papers in journals and 6 International conference proceedings.