Article can be accessed online at http://www.publishingindia.com

Efficient Security for Data Communication in Wireless Sensor Networks using Raspberry PI

Shruti V Jadhav*, Ajay Acharya**

Abstract

Information technology finds its wide application in many fields like healthcare, border security etc. where security of information is very important. With advancement in wireless sensor networks, gathering and processing of data have become much easier and affordable, due to its wide application, providing security to data is required. This work proposes a prototype that uses Raspberry Pi as secure sensor node that performs both regular encryption and XML encryption to provide security to the data collected by sensor nodes and transmit it to the master node in the WSN. The design uses a temperature sensor which can be replaced by any other sensor going forward and can be used in the field of healthcare, border security, and many other areas. The data are collected, processed, encrypted, and wirelessly communicated to the master node. Since there is direct interfacing between the sensor and Raspberry Pi, there is no risk of data loss. The idea of this paper is to meet the goal of comparing the two encryption techniques in terms of processing time and number of characters and to see which is better in terms of both the parameters.

Keywords: Wireless Sensor Networks, Temperature-sensor, Raspberry Pi, XML Encryption

Introduction

Wireless Sensor Node (WSN) consists of spatially distributed sensors that monitor multiple conditions of the environment like temperature, pressure, sound etc. These sensors pass the data collected cooperatively to the main central location where the data is stored and maintained. Wireless sensor network is made up of number of nodes ranging from several hundreds to thousands. The various functions that a sensor node has in wireless sensor network include processing, gathering sensory information, and communicating with other nodes in the network.

Raspberry Pi is a small, cheap, education oriented device which has a same functionality as that of a computer. It is a small credit card sized computer which was introduced in 2012. It shows very good performance and very well capable of being interfaced with many other devices. (Raspberry Pi, 2015)

Fig. 1: Raspberry Pi (Raspberry Pi, 2015)

Raspberry Pi is similar to other computers which makes use of an operating system called Raspbian. Raspbian is a flavour of Linux and since it is open source, the price of the platform is very low which makes it a best and a great match for Raspberry Pi (Vujovic & Maksimovic, 2014). The main advantages of using Raspbian is that it provides an easy transition for the ones that are not familiar with

* KLE Dr MSSCET, Belgaum, Karnataka, India.
** KLE Dr MSSCET, Belgaum, Karnataka, India.

Linux command line since it has a desktop environment which is similar to windows called Light Weight X11 Desktop Environment. Raspbian comes with ready software that is useful for writing programs. It performs the compilation using on-chip floating point calculations which is faster than the software based method. Also this operating system finds a wide spread community support.

The main goal of this work is to provide a prototype system that makes use of Raspberry Pi as a sensor node to which a temperature sensor is attached and is directly interfaced with the Raspberry Pi. The Raspberry Pi performs both regular encryption as well as XML encryption to encrypt the temperature values and communicate the encrypted data securely through wireless communication to a central location or master node where the values are stored for further reference.

Also comparative analysis is performed between regular encryption and XML encryption in terms of the time taken and bandwidth usage. Hence the overall idea is to provide an efficient security for the data communication in wireless sensor network in terms of processing time and number of characters.

Related Work

Wireless sensor networks consists of number of sensor nodes distributed over specific area and the main aim is to collect large amount of data over wide spread location (Vujovic & Maksimovic, 2014). For collecting large amount of data we need more sensor nodes and deploying more number of sensor nodes needs to be in budget. The size and cost of the sensor node play an important role in the deployment of these sensor nodes. Lesser the cost and size of the sensor nodes, more number of sensor nodes can be deployed and hence more data can be collected. A reduction in per-node cost will result in the ability to purchase more nodes, to deploy a collection network with higher density, and to collect more data. The comparison of size, weight and cost of basic model of Raspberry Pi and the other popular wireless sensor nodes is given in Table 1.

The values presented in Table 1 show Raspberry Pi's advantage against other systems lies in its smallest per unit price.

Maurya and Shukla (2013) explain the emerging technology of wireless sensor networks and also investigate

Table 1: Comparison of Size, Weight and Cost

Name	*Size (mm)*	*Weight (g)*	*Cost per node US$*
Raspberry Pi	85.6*53.98*17	45	25-35
MicaZ	58*32*7	18	99
TelosB	65*31*6	23	99
Iris	58*32*7	18	115
Cricket	58*32*7	18	225
Lotus	76*34*7	18	300

certain aspects which are important towards the performance of the sensor nodes. Here comparison is made of processor and memory of various sensor nodes and comparing these with Raspberry Pi shows the advantages of Raspberry Pi over the other sensor nodes. This work also focuses on certain disadvantages of these typical sensor nodes like less processing power, threatened by less life span etc. all these can be overcome by making use of a Raspberry Pi as a sensor node. Table 2 summarizes the comparison of processor and RAM of Raspberry Pi with other sensor nodes.

Table 2: Comparison of CPU and Memory

Name	*Processor*	*1*
Raspberry Pi	ARM BCM2835	512 M
MicaZ	ATMEGA128	4 K
TelosB	TI MSP430	10 K
Iris	ATMEGA1281	8 K
Cricket	ATMEL128L	4 K

Wireless sensor network consists of many tiny sensor nodes. It is becoming a trending technology these decades. These sensor nodes do not have access to renewable sources of energy. There are many challenges in this field (Kalita & Kar, 2009). This paper explains many various threats in wireless sensor networks. The threats vary according to various network layers and this work proposes some measures to deal with these threats or attacks. One such attack is tampering which can be dealt by making use of encryption, where normal data is changed to an unreadable form with help of some encryption technique.

Panda (2014) summarizes the security in wireless sensor networks using cryptographic techniques; however the wireless sensor networks have many limitations such as computational speed, limited memory transmission range,

unreliable communication, unreliable transfer and most of all they are deployed in an unattended environment where it is more prone to attacks than usual. Hence cryptographic techniques can be used to prevent such attacks to some extents.

Anand, Chandrakanth and Giriprasad (2012) and Alajmi (2014) have briefed about various attacks on thewireless sensor networks. In today's world where wireless sensor networks is a blooming technology it finds its applications in various fields like healthcare, military, defence etc. where providing security becomes more essential. This paper explain various attacks such as node capture, denial of service attack etc. One such issue discussed here is tampering, which can be addressed by taking necessary measures and hence use of cryptography in such cases is indispensable.

(W3C Recommendation, 2014) XML is spreading quickly as a format for electronic documents and messages. As a consequence, greater importance is being placed on the XML security technology. Against this background research and development efforts into XML security are being energetically pursued (Miyauchi, 2005).

XML encryption technology is used for data encryption. After the encryption process XML formatted data is generated, reassembled in XML format data, and then this encrypted XML data is sent to one or more receivers. The main goal of XML encryption is to use the XML file element and content whose purpose is to make sure the data confidentiality and integrity of data storage and exchange.

Proposed Work

Nowadays wireless sensors find its applications in many fields like home automation, military and defence, meteorological monitoring, intelligent transportation etc. In some applications security is very crucial hence the main goal of this project is to find out a way to secure the data and also to find the way that is efficient.

This project implements a prototype system that makes use of Raspberry Pi as a sensor node, to which a temperature sensor is attached and directly interfaced with the Raspberry Pi. The Raspberry Pi performs both regular encryption as well as XML encryption to encrypt the temperature values and communicate the encrypted data securely through wireless communication to a central location or master node, where the values are stored for further reference.

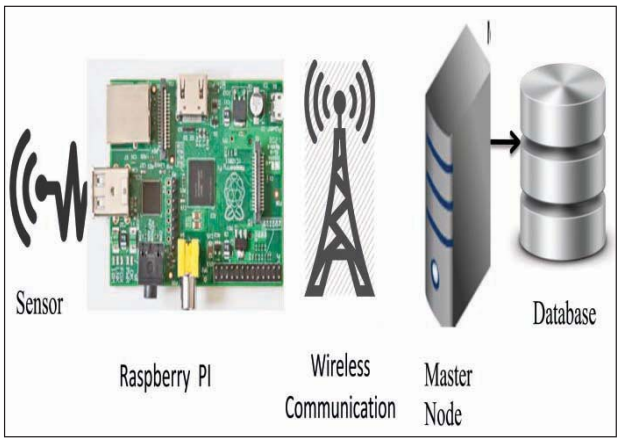
Also comparative analysis is performed between regular encryption and XML encryption in terms of the time taken to encrypt and bandwidth usage. Hence the overall idea is to provide an efficient security for the data communication in wireless sensor network in terms of number of characters and the time taken or the processing power.

The encrypted values are wirelessly communicated to the master node where the values are decrypted and stored in the master nodes database for further reference.

System Design

The architectural design consists of a sensor directly interfaced with Raspberry Pi which acts as a sensor node and the data read by the sensor node is encrypted using two encryption techniques: regular encryption and XML encryption.

Fig. 2: System Architecture



Implementation

The implementation involves the following steps which describe the process in detail

Step 1: Set up Raspberry Pi

The Raspberry Pi uses a SD card for storage. When we first power up the Raspberry Pi we need to select the operating system and configure it. A static IP address is

given to the Raspberry Pi which is used to remotely login to it using software Putty.

Step 2: Interface the temperature sensor with Raspberry Pi

Temperature sensor used is a one wired digital temperature sensor DS18B20. It has 3 pins +5V, ground and output. Interfacing temperature sensor involves connecting the sensor to correct pins on the Raspberry Pi.

Step 3: Initialize the temperature sensor

Before using the temperature sensor it needs to be initialized. A file is written named ds18b20.sh which contains all the commands necessary to initialize the temperature sensor. This file must be executed using command ./ds18b20.sh

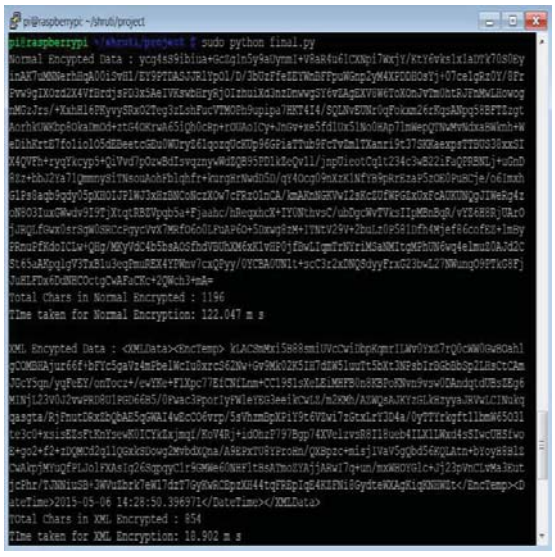
Step 4: Read the temperature values on to the Raspberry Pi

The temperature values are continuously read into the Raspberry Pi

Step 5: Encrypt the values using regular encryption and XML encryption

The read parameter values are encrypted using regular encryption technique. AES encryption algorithm is used to encrypt the temperature, date and time.

Fig. 3: XML Encrypted Data



Also the same parameters are encrypted using XML encryption. XML encryption provides the flexibility of partial encryption. Here in our prototype system temperature is crucial to us so only this is encrypted and the date and time are sent as plain text.

Step 6: Calculate the number of characters and time taken to encrypt in both XML and regular encryption technique

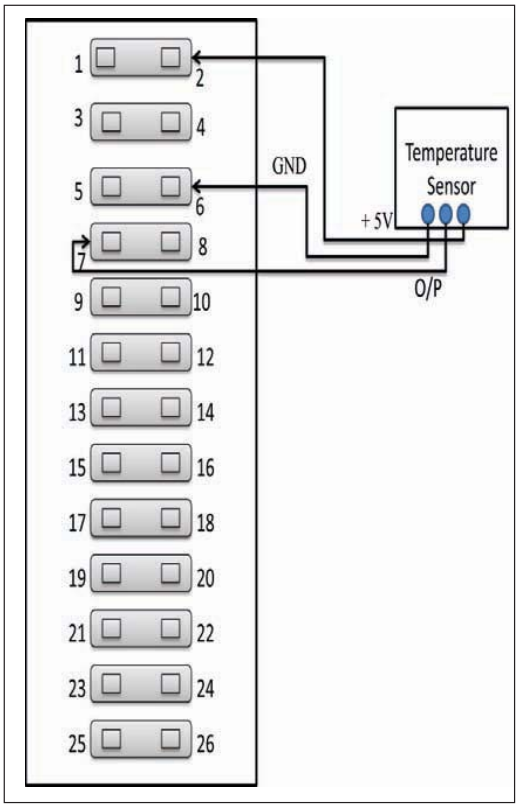
Step 7: Wirelessly communicate the encrypted values to master node.

Step 8: Master node decrypts the values and stores in the database

This step is implemented to decrypt the temperature and store in the database. VB.Net application is also running on the master node that displays graphs of XML versus regular, real time temperature.

The temperature sensor used is DS18b20. The DS18B20 is a digital thermometer. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range

Fig. 4: Pin Connection of Temperature Sensor



of -10°C to +85°C. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

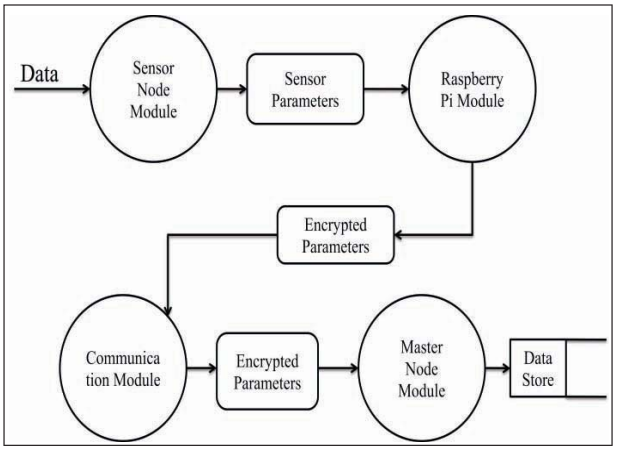
The pin connections to connect the temperature sensor to Raspberry Pi are shown in Fig.4. The +5V is connected to pin no. 2, Ground is connected to pin no. 6, and the output to pin no. 7.

First a file ds18b20.sh is to be written which contains the commands required to initialize the temperature sensor. The command used on Raspberry Pi to initialize the temperature sensor is ./ds18b20.sh

Regular Encryption

The temperature is read along with the current date and time into the Raspberry Pi and using AES encryption algorithm encryption takes place. Any other encryption algorithm like DES, 3DES, RSA can also be used. The algorithm encrypts all the 3 parameters including temperature, date, time, and the time taken to encrypt and also the characters in cipher text are calculated.

Fig. 5: Regular Encryption



XML Encryption

Next the same data is passed through XML encryption technique which provides us with the flexibility of encrypting only the values that are crucial to us. Here in our prototype system the temperature value is crucial to us, which is why the temperature is encrypted, all the other data i.e. date and time are sent as plain text resulting in less usage of bandwidth.

Step 1: Identify XML element or element content to be encrypted

Step 2: Select the encryption method

- This includes the encryption algorithms, keys etc.

Step 3: Encrypt the data

Step 4: Encode the result

Step 5: Replace the cipher text back into the XML

Fig. 6: Flow Chart

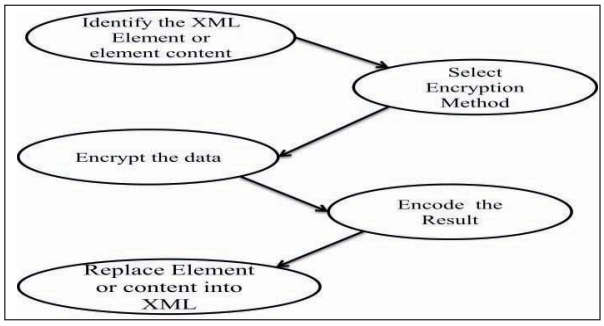
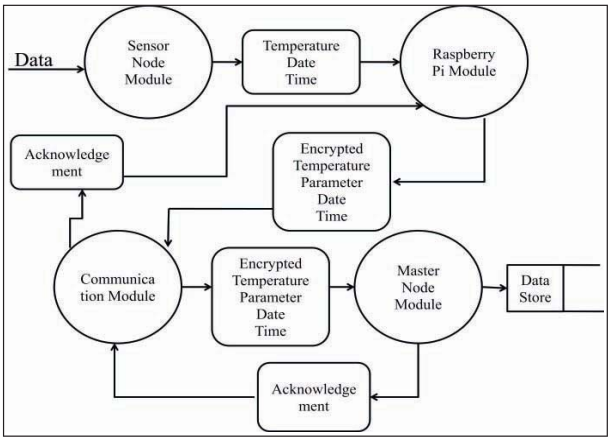


Fig. 7: XML Encryption

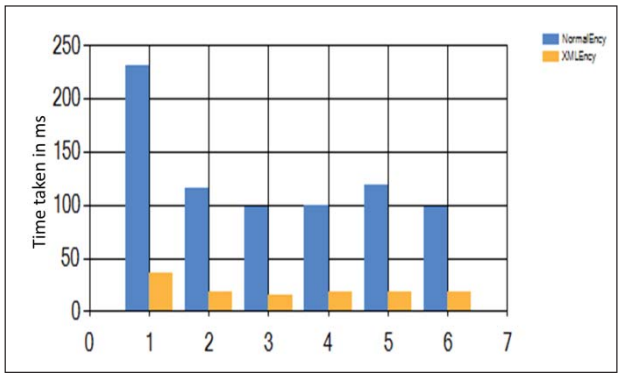


Results

After implementing this project the results found are shown in Graph 1.

The graph shows that the processing time in XML encryption is way better compared to the regular encryption.

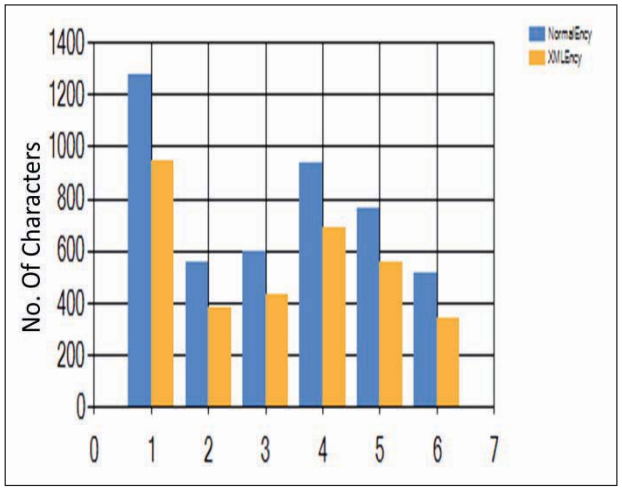
Graph 1: Processing Time in Regular and XML Encryption



- The time taken to process in XML is better compared to regular encryption
- Since sensor nodes are very tiny and have less processing power, hence low processing time is always an advantage.

From the results in Graph 2, it is clear that number of characters after regular encryption is more and in XML encryption it is less.

Graph 2: Number of Characters in Regular and XML Encryption



Conclusion

Using Raspberry Pi instead of a typical wireless sensor node has a few advantages. It allows providing more security because of its good processing power in comparison with the other sensor nodes. In today's world where we all are more keen on using both the bandwidth and processing time judiciously, XML encryption technique provides the flexibility of partial encryption and also is the best suited one because it uses less bandwidth and also less processing time compared to regular encryption technique. Since the sensor nodes are tiny it is beneficial to have low processing time. So XML encryption can be used as an efficient security method to be provided for data communication in Wireless Sensor Network (WSN) by making use of Raspberry Pi as a sensor node.

References

Alajmi, N. (2014). Wireless sensor networks attacks and solutions. *International Journal of Computer Science and Information Security*, July, 12(7), 1-4.

Anand, D. G., Chandrakanth, H. G., & Giriprasad, M. N. (2012). Security threats & issues in wireless sensor networks. *International Journal of Engineering Research and Applications,* 2(1), 911-916.

Kalita, H. K., & Kar, A. (2009). Wireless sensor network security analysis. *International Journal of Next-Generation Networks*, 1(1), 1-10.

Maurya, M., & Shukla, S. R. (2013). Current wireless sensor nodes (Motes): Performance metrics and constraints. *International Journal of Advanced Research in Electronics and Communication Engineering*, 2(1), 0-45.

Miyauchi, K. (2005). XML signature/encryption- The basis of web service security. *NEC Journal of Advanced Technology*, 2(1), 35-39.

Panda, M. (2014). Security in wireless sensor networks using cryptographic techniques. *American Journal of Engineering Research*, 3, 50-56.

Raspberry Pi. (2015). *What is Raspberry Pi?* Retrieved from https://www.raspberrypi.org/community/

Vujovic, V., & Maksimovic, M. (2014). *Raspberry Pi as a Wireless Sensor Node: Performances and constraints.* In 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), (pp. 1013-1018).

W3C Recommendation (11 April 2013). *XML Encryption Syntax and Processing Version 1.1.* Retrieved fromhttp://www.w3.org/TR/xmlenc-core1/