

Tunnelling Authenticated Data from Raspberry PI and Message Intimation via GSM

S.Divya

Assistant Professor, Department of CSE, RVS Faculty of Engineering, Coimbatore, Tamilnadu, India.

Abstract –A credit card sized single board computer named raspberry pi which is built using ARM11 processor that plugs into your TV or any display and a keyboard. The paper deals with a system which maintains the attendance records automatically. This system comprises of a fingerprint reader to manage the attendance of the person. The attendance generated is stored in the database and the information is passed via SSH tunnel to the server system. Also the attendance report is sent as message to the person's mobile periodically. This finger print system is secured in maintaining the attendance in any organization. The recorded finger print detail is shown on the display of raspberry pi. Also a GSM/GPRS module is used to send intimation to person's mobile about the attendance data stored in server.

Index Terms – Fingerprint, GPRS, GSM, ARM11 processor

1. INTRODUCTION

Raspberry Pi is a credit card sized single board computer, designed with the intention of promoting the teaching of basic computer science in schools. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games. It also plays high-definition video. We want to see it being used by kids all over the world to learn programming. It is built using ARM11 processor.

This device does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage. A fingerprint in its narrow sense is an impression left by the friction ridges of a human finger. The recovery of fingerprints from a crime scene is an important method of forensic science.

The rest of the paper is organized as follows. Related works are explained in section 2. System Analysis is explained in section 3. Architecture of the System is explained in Section 4 and Concluding remarks and Future enhancement is given in section 5.

2. RELATED WORK

Many related works regarding attendance management in schools, colleges and industries exists:

In [1], the attendance of each students are taken during both the examination times and also on regular hours. A fingerprint device is used and the attendance of the students are stored in

the database. Authors in [2, 3] developed a college attendance system using wireless ZigBee technology. The attendance report will be sent to the respective department HOD or class in-charge once in 15 days. The report can also be sent to the parents e-mail id. Some also used RFID to track attendance of the students.

Authors in [4] designed a student metric card embedded with an RFID tag for tracking their whereabouts. When the metric card passes through the RFID reader, it will trigger the system to read the data from the RFID tag to the database where the access data can be viewed online by the management for monitoring purposes.

In [5] an attendance management system has been created mainly for employees. It records the attendance of employees once per day. Wireless attendance management system are also designed by some authors.

In [6], attendance is managed wirelessly. Once the presence for a particular employee is taken through a biometric device, the presence is stored in the database through Wi-Fi. Authors in [7] created an attendance management system for students using a fingerprint sensor.

3. SYSTEM DESIGN

3.1. Existing System

3.1.1. Working of module :

The module consist of raspberry pi which is a credit card sized computer connected to finger print reader with arduino board and LCD display. This pi can be connected to any kind of displays. Here an LCD display is used to view the finger print details of the person. Each person has a unique finger print id. The finger print id are stored in the database of the server machine. When finger print reader reads a finger print, it matches with the list of ids available in the server machine database. Once when the id is matches then the person's id, name, date, time of reading is shown in the LCD screen and recorded in the local pi's database. Otherwise, an error message will be shown on the screen. Figure 1 shows raspberry pi

Each entry is recorded in the database of raspberry pi and sent to the database of the server machine through a LAN

connection. The user or the person is been authenticated from the reading process itself.



Figure 1 Raspberry Pi

3.1.1 SSH connection :

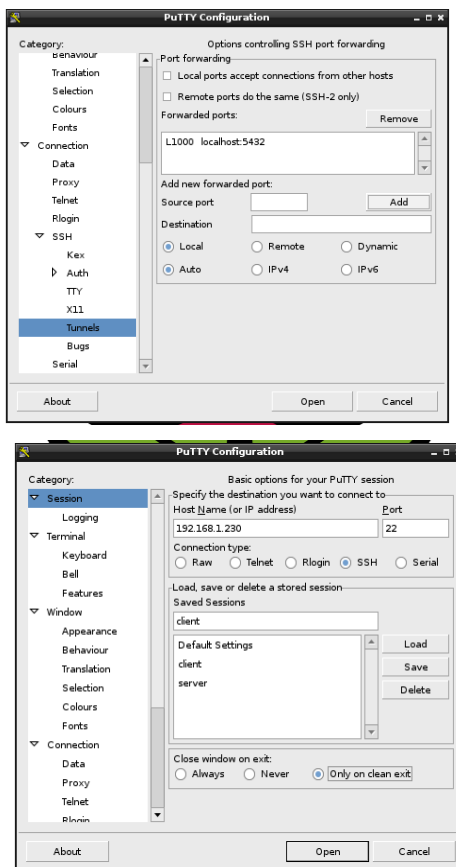


Figure 2 SSH Tunnel between client and server

Secure Shell (SSH) is a UNIX-based command interface and protocol for securely getting access to a remote computer. SSH is actually a suite of three utilities - slogin, ssh, and scp - that are secure versions of the earlier UNIX utilities, rlogin, rsh, and rcp. SSH commands are encrypted and secure in several ways. Both ends of the client/server connection are

authenticated using a digital certificate, and passwords are protected by being encrypted. SSH allows you to connect to your server securely and perform Linux command-line operations. SSH tunnel is created between the client (raspberry pi) and the server machine using putty as shown in figure 2.

3.1.3 pgAdmin III :

pgAdmin III is a comprehensive PostgreSQL database design and management system for Unix and Windows systems. It is freely available under the terms of the PostgreSQL License and may be redistributed provided the terms of the license are adhered to. The project is managed by the pgAdmin Development Team. This software was written as a successor to the original pgAdmin and pgAdmin II products, which though popular, had limitations in the design that prevented them being taken to the 'next level'. pgAdmin III is written in C++ and uses the excellent wx Widgets (formerly wxWindows) cross platform toolkit. Connection to PostgreSQL is made using the native libpq library. Figure 3 shows the pgAdmin window. Here pgAdmin III is used to maintain the backup of the database. In case of any failure of connection or lose of data, then immediately the server can restore the database and send it through LAN to client machine.

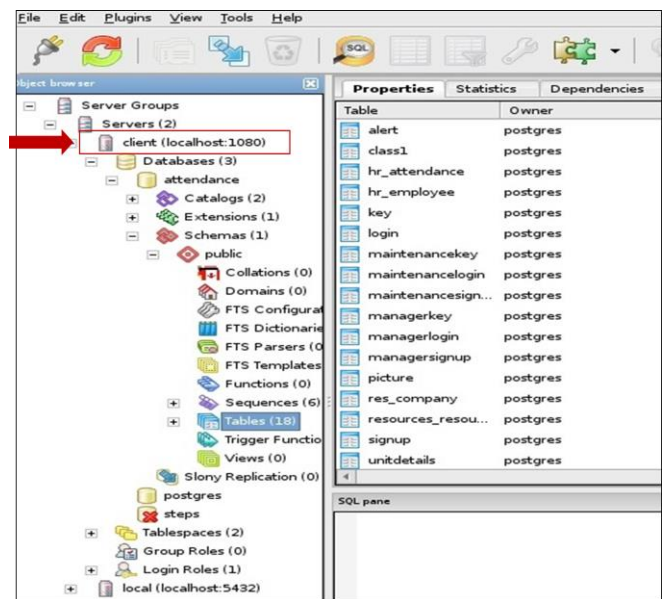


Figure 3 pgAdmin window.

PostgreSQL manages concurrency through a system known as multiversion concurrency control (MVCC), which gives each transaction a "snapshot" of the database, allowing changes to be made without being visible to other transactions until the changes are committed. This largely eliminates the need for read locks, and ensures the database maintains the ACID (atomicity, consistency, isolation, durability)

principles in an efficient manner. PostgreSQL offers three levels of transaction isolation: Read Committed, Repeatable Read and Serializable. Because PostgreSQL is immune to dirty reads, requesting a Read Uncommitted transaction isolation level provides read committed instead.

In this database, all objects (with the exception of roles and table spaces) are held within a schema. Schemas effectively act like namespaces here, allowing objects of the same name to co-exist in the same database. Schemas are analogous to directories in a file system, except that they cannot be nested, nor is it possible to create a "symbolic link" pointing to another schema or object.

4. PROPOSED MODELLING

3.2.1 GSM/GPRS module :

The GSM/GPRS module is connected to the client machine where the attendance record of a person is sent as a message to their phone number periodically. The message contains the unique id, name, total working days, etc. GSM module shown in figure 4 sends message information from the server machine database.

For AT commands:

- Use capital letters for AT commands.
- Send CR (Carriage return) and LF (Line feed) after the AT command.
- Place the serial communication jumpers in the right position.
- Use an external power supply and place the power jumpers in the right position. If the shield is powered from the Raspberry Pi, the power jumper must be in Arduino 5V position. If the shield is powered from the Vin input (in the shield), the power jumper must be in next position.

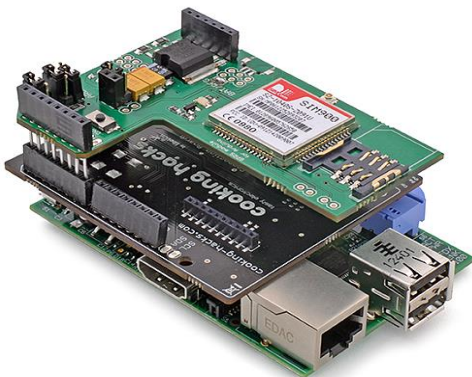


Figure 4 GSM/GPRS module

Basic configuration:

Connect the GPRS/GSM shield to the Raspberry Pi to Arduino shield. Then introduce the SIM card in the GPRS/GSM shield. Then open on the Raspberry a serial port terminal to communicate with the GPRS/GSM shield (just run cutecom in a terminal). This step can be done directly on the Raspberry with a keyboard, mouse and screen plugged, or via ssh or on the sever machine.

5. WORKING AND ARCHITECTURE

The architecture comprises of raspberry pi connected to finger print reader with arduino board, GSM/GPRS module and LCD display. Raspberry pi is connected through local area connection (LAN) with server machine. The server machine can be of any operating system enabled with SSH connection. SSH tunnel is created between the client and the server machine to have secure communication between the two systems. The finger print image is stored in binary format blob data type of 0s and 1s in the database. Database fields are finger print id, finger print image, date, time, name of the person, type of work. The client machine and server machine contains postgresql database with pgAdmin as a backup and restore platform.

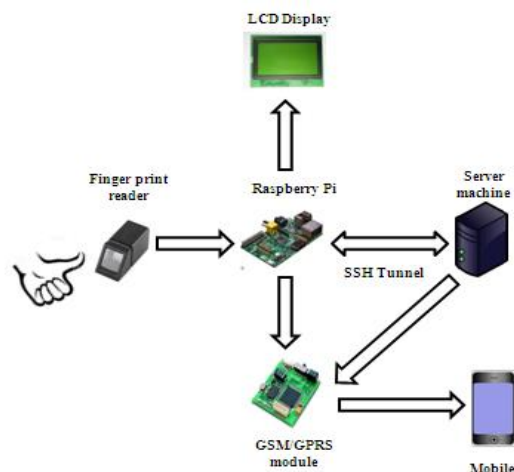


Figure 5 Architecture of attendance recorder system

6. CONCLUSION AND FUTURE ENHANCEMENT

This paper comprises of the fingerprint authentication using raspberry pi for an attendance management system for the purpose of maintaining attendance details of a person. The client and server is connected via SSH and the database values are sent from client to server. In case of any lose of data the server restores the data using pgAdmin and sends the data to the client machine. GSM/GPRS module is used here to send the attendance details as a message to the person's phone.

In future the work can be extended for maintaining the attendance for students in an institution using the radio frequency identification and also this same work can be extended to face detected to maintain the attendance.

REFERENCES

- [1] O. Shoewu and O.A. Idowu (2012), "Development of Attendance Management System using Biometrics", Pacific Journal of Science and Technology, Volume 13. Number 1.
- [2] Gunjan Talaviya, Rahul Ramteke and A.K.Shete (2013), "Wireless Fingerprint Based College Attendance System Using ZigBee Technology", International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume-2, Issue-3.
- [3] Zhao, Cheah Wai, Jayanand Jegatheesan, and Son Chee Loon. (2015) "Exploring IOT Application Using Raspberry Pi." International Journal of Computer Networks and Applications (IJCNA), 2(1), 27-34.
- [4] Herdawatie Abdul Kadir, Mohd. Helmy Abd. Wahab, Zarina Tukiran Mohd Razali Mohd Tomari and Mohd Norzali Hj. Mohd (2010), "Fusion of Radio Frequency Identification (RFID) and Fingerprint in Boarding School Monitoring System (BoSS)", ISBN 978-953-7619-74-9, pp. 356.
- [5] Seema Rao and Prof.K.J.Satoa (2013), "An Attendance Monitoring System Using Biometrics Authentication", Volume 3, Issue 4 ISSN: 2277 128X.
- [6] Yongqiang Zhang and Ji Liu (2007), "Wireless Fingerprint Attendance Management System", Proceedings of the 2007 WSEAS International Conference on Computer Engineering and Applications.
- [7] Tabassam Nawaz, Saim Pervaiz, Arash Korrani, Azhar-Ud-Din (2009), "Development of Academic Attendance Monitoring System Using Fingerprint Identification", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.5.
- [8] <https://www.sparkfun.com/products/11792>
- [9] <https://www.evernote.com/shard/s325/sh/b135be0c-954b-40c9-93c7-f88886536a5/f177bfc6bac114b2dc70d1113195ae04>
- [10] <http://www.raspberrypi.org/forums/viewtopic.php?f=41&t=63013>
- [11] http://www.rippublication.com/irph/ijict_spl/11_ijictv3n5spl.pdf
- [12] <http://arduino.cc/en/Main/ArduinoWiFiShield>
- [13] <http://arduino.cc/en/Main/ArduinoGSMShield>
- [14] <https://projects.drogon.net/raspberry-pi/wiringpi/lcd-library/>
- [15] <http://www.cooking-hacks.com/documentation/tutorials/raspberry-pi-gprs-gsm-quadbandsim900http://www.raspberrypi.org/forums/viewtopic.php?f=44&t=69059>
- [16] <http://raspberrypi.stackexchange.com/questions/7702/raspberry-pi-and-fingerprint-scanner>
- [17] <http://www.rackspace.com/knowledge-centre/article/mysql-connect-to-your-database-remotely>
- [18] <http://www.cyberciti.biz/tips/how-do-i-enable-remote-access-to-mysql-database/>
- [19] <http://www.raspberrypi.org/phpBB3/viewtopic.php?f=34&t=12000>
- [20] <http://www.raspberrypi.org/phpBB3/viewtopic.php?f=37340&t=317829>
- [21] <http://www.thegeekstuff.com/2010/08/allow-mysql-client-connection/>
- [22] <http://www.cyberciti.biz/tips/how-do-i-enable-remote-access-to-mysql-database/>
- [23] <http://www.raspberrypi.org/phpBB3/viewtopic.php?f=34&t=12000>
- [24] <http://www.raspberrypi.org/phpBB3/viewtopic.php?f=37340&t=31782>
- [25] <http://www.thegeekstuff.com/2010/08/allow-mysql-client-connection/>
- [26] <http://www.satsignal.eu/ntp/Raspberry-Pi-NTP.html>