An Implementation of MD5 Hash Algorithm for RFID Tags

Nurbek Saparkhojayev
MSc in CSCE, PhD candidate, Senior Lecturer,
Suleyman Demirel University, Almaty, Kazakhstan
Kaskelen, Almaty, Kazakhstan

Olzhas Shaiken
Bachelor student,
Suleyman Demirel University, Almaty, Kazakhstan
Kaskelen, Almaty, Kazakhstan
shaiken.olzhas@gmail.com

ABSTRACT
This paper discusses RFID technology, and shows the implementation of MD5 hash algorithm for RFID tags to protect private data from counterfeiting and copying. Privacy and security of RFID tags are discussed and the implementation of this algorithm is shown. Authors present new technology of reading the data from RFID- cards by using MD5 hash function. The reason of applying this algorithm is trivial: it is fast comparing to other algorithms, and can handle small-fixed size of data, which is stored in low-cost RFID-cards, which means that it has 1 Kb of memory. This work was done as a part of project called "Smart Campus" in Suleyman Demirel University, Almaty, Kazakhstan.

General Terms
RFID Privacy and Security, Algorithms.

Keywords
RFID technology, counterfeiting, MD5 algorithm, privacy, security.

1. INTRODUCTION
Radio-frequency identification (RFID) is a technology that uses radio waves to transfer data from an electronic tag – called an RFID tag or label, which is attached to an object – through a reader for the purpose of identifying and tracking the object. It is possible for some RFID tags to read from several meters away and beyond the line of sight of the reader. Nowadays, RFID systems have been widely used in many different application areas of industry, such as: product tracking through manufacturing and assembly , control of inventory, parking lot access and control, container tracking, ID badges and access control, equipment tracking in hospitals, etc[1]. Compared to other automatic identification technologies, such as optical barcode systems, RFID-technology has several advantages. The data located in tag can be read automatically beyond the line of sight, thru certain materials, and from a range of several meters [2]. According to its usage, RFID tag can have different types. There are several frequencies are available, including LF, HF, UHF, and microwave. Depending on the country in which the RFID tag is used, these frequencies may vary. In this project, LF-based RFID tag was used.

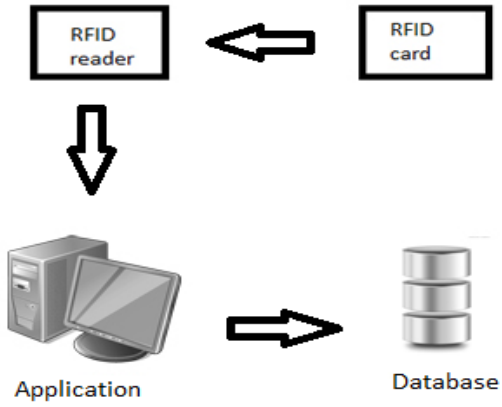

Fig 1: RFID technology workflow

Fig 2: RFID tags classification

In [3] research paper, authors have implemented system for automatic monitor of students' attendance based on RFID- technology. They demonstrated how to automate an entire student-attendance registration system within an educational institution by putting together the architecture and prototype of a RFID system transmitted over Ethernet. In [4], researchers showed different approach for attendance checking system. They designed and implemented wireless

attendance management system based on iris recognition. In [1], the beta-version of "Smart Campus" project was presented. They presented first module, which aimed to automatically check students' attendance. However, they built their system without using an MD5 algorithm; the most important concern was to start this project. In this research paper introduced the implementation of MD5 for this project. Also, there was some other research work done [5] in Europe, in which authors proposed attendance management system by the use of computer vision algorithms. They used real-time face detection algorithms integrated on an existing Learning Management System (LMS), which automatically checks, detects and registers student attending on a lecture. This is a quite interesting approach since this software works with almost no-human interaction. MD5 is well-known and most widely used cryptographic hash function at current time. MD5 is the hash function designed by Ron Rivest as a strengthened version of MD4 [6]. It has been the most widely used secure hash algorithm, particularly in Internet-standard message authentication. MD5 has also been proposed as the default authentication option in IPv6 [7]. In this research, authors relied on a MIFARE RFID-tag, specifically, the MIFARE MF1ICS50 typed RFID-tag. This type of tag was developed by NXP to be used in a contactless smart card according to ISO/IEC 14443 Type A. The MIFARE MF1ICS50 IC is used in such applications as public transportation ticketing, which major cities of the world have adopted as their e-ticketing solution. The MF1ICS50 chip consists of a 1 K-byte EEPROM, a RF-Interface and a Digital Control Unit. Energy and data are transferred via an antenna comprising a coil with a few turns directly connected to the MF1ICS50 [8].

As RFID-reader, Stronglink's SL040A was used due to its cheapness and ease of use. This reader enables the contact-free reading and writing of operations and works on a 13.56 MHz frequency. It can read the unique serial number of all MIFARE cards, including MIFARE Classic 1K, MIFARE Classic 4K, MIFARE Mini, Ultralight, DESFire, MIFARE Plus, MIFARE ProX, etc.[9].

Fig 3: RFID reader SL040A

2. IMPLEMENTATION

Program reads from RFID card and checks for authenticity. Each RFID card has 4 blocks and 16 sectors, in which data is stored. Size of each sector is 32 digits long, because of this, all data will be 32 length. RFID cards have their own 32 digits long CardID stored in zero block, zero sector. Beside this CardID, another ID was created, which is called UniqueID, and it is 32 digits long. These two different ID's are stored in .txt file. Then, after combining these both IDs by XOR operation for each RFID card, the new ID so-called SuperID is created, and the size of this SuperID is the same as previous ones- 32 digits long.

$$CardID \oplus UniqueID = E_{MD5}(SuperID)$$

The new created SuperID is stored in database, and all SuperID's are encrypted by MD5 algorithm to protect the data, preventing hacker from possible threats such as reading the data, copying the SuperID, and so on. Nevertheless to the fact that MD5 algorithm is known as weak algorithm, this algorithm was chosen because its output is 128 bits and 32 digits long, and comparing to algorithms like SHA-512, MD6, and others it is the optimal one since in RFID cards which were used for this project, only 32 digits long outputs can be produced. As it was said previously, each SuperID is stored in database with other important info such as Student's full name, Faculty, etc. SuperID's are stored in zero block of first sector of RFID cards.

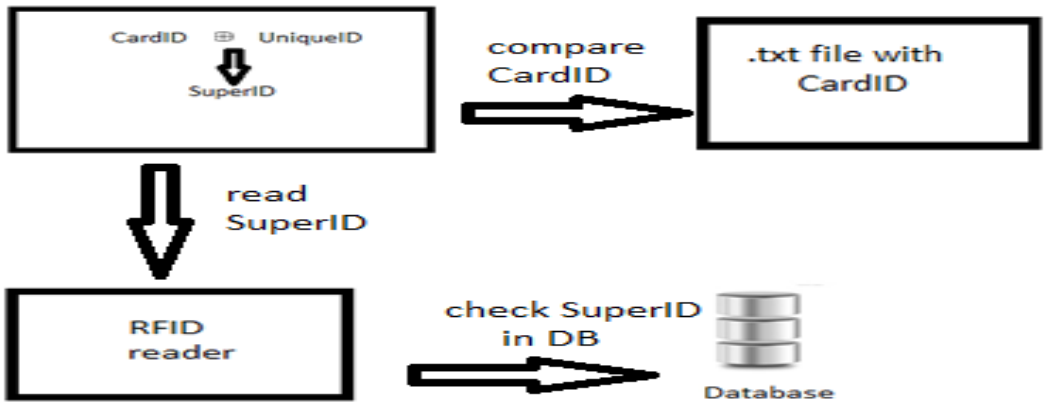

Fig 4: The workflow of algorithm, implemented in this research

The overall structure of this workflow:

1. Student comes to the reader and processes his/her own card.

2. Reader reads SuperID and checks in database this value with stored one. If there is no such SuperID, program prompts "Attempt of copy or not registered card" and exits.

3. After successfully passing SuperID, system checks CardID. In the case of copying SuperID and its use, the system will detect by checking CardID, which is unique.

4. If everything is ok with this CardID, then system displays the following output:

5. After this step, system will authorize cardholder.

3. CONCLUSION AND FUTURE WORK

The authors of this research paper have shown how MD5 hash algorithm can be used for RFID-cards. Firstly, this algorithm will be used as an official algorithm for the project called "Smart Campus". This project consists of multiple independent modules, such as "Door Control System", "Attendance Checking System", "Staff Attendance Monitoring System", "Library System", and others. The cards that have been employed for this specific system are RFID-cards, and the algorithm used has shown stable and reliable results; moreover, this algorithm has secured important data that have been stored on these cards. It has been planned that these RFID-enabled cards can be put to use at the university and may replace student ID cards. Personnel and students, alike, can use these cards for many purposes; additional functions can always be incorporated into the system and greater security provided to the cards. RFID- technology continues to develop, and the time has come for us to avail ourselves of its promise and convenience. The main aim of this research has been to demonstrate an algorithm which is based on MD5 hash function, and build a system reliant on it.

For the future work, this algorithm will be re-tested for some time to check its stability in real-time. At the meantime, authors are planning to check other algorithms and make a case study to choose the best one. But, for now, this algorithm will be used at the beginning. There was research done in [10], which showed how to build and implement Library Management system based on RFID. Simultaneously, other types of RFID-cards should be checked and the best one should be picked, which should have enough memory size so that can be kept more data inside of them and to adopt other algorithms which work with long length outputs. Furthermore, the possibility of adding some mobile tools like GPS, GSM and so on is considered, and the project for implementing such a system is started. Future plan is to use GPS and GSM technologies in educational system, and the work that was done in [11] is an impulse for this project implementation.

4. ACKNOWLEDGMENTS

Special thanks are given to Suleyman Demirel University, Kaskelen, Almaty,Kazakhstan for providing us with all necessary equipment and for their patience. Last but not least thanks are given to the research department of the University of Technology (em. Niyazi Ari, Prof. Dr. sch. Techn. ETH) Zurich, Switzerland.

Appendix

```
import java.awt.FlowLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;
```

```
import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.InputStreamReader;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Timer;
import java.util.TimerTask;
import java.util.logging.Handler;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;

/**
*
* @author shaiken
*/
public class RFID extends JFrame implements Runnable{
static JFrame frame;
JButton button;
JTextField passfield;
JLabel label;
Box box1, box;
public Connection connection;
Statement statement;
ResultSet resultSet;
ArrayList<String> keys;
ArrayList<String> card_id;
Timer timer;
Timer timer2;
Thread timer3;
TimerTask timerr;
String qq = null;
ImageIcon icon;

public String getMD5(String input) {
   try {
      MessageDigest md =
MessageDigest.getInstance("MD5");
      byte[] messageDigest = md.digest(input.getBytes());
      BigInteger number = new BigInteger(1, messageDigest);
      String hashtext = number.toString(16);

      while (hashtext.length() < 32) {
      hashtext = "0" + hashtext;
      }
      return hashtext;
      }
      catch (NoSuchAlgorithmException e) {
      throw new RuntimeException(e);
      }
}

public RFID() throws Exception {
   Class.forName("com.mysql.jdbc.Driver");
   String dbURL = "jdbc:mysql://localhost/rfid";
   String user = "root";
   String pass = "";
   this.connection = DriverManager.getConnection(dbURL,
user,pass);
   FileInputStream finstream = new
FileInputStream("card_id.txt");
   DataInputStream dinstream = new
DataInputStream(finstream);
   BufferedReader breader = new BufferedReader(new
```

```
   InputStreamReader(dinstream));
   String strLine;
   card_id = new ArrayList<String>();
   while ((strLine = breader.readLine()) != null) {
      card_id.add(strLine);
   }
   String card11 = card_id.get(0);
   System.out.println(card11);
   BigInteger i = new BigInteger(card11,16);
   System.out.println(i);
   String card1= i.toString();
   String card2 = card_id.get(1);

   FileInputStream finstream1 = new
FileInputStream("unique_id.txt");
   DataInputStream binstream1 = new
DataInputStream(finstream1);
   BufferedReader br1 = new BufferedReader(new
InputStreamReader(binstream1));
   String strLine1;
   ArrayList<String> unique_id = new ArrayList<String>();
   while ((strLine1 = br1.readLine()) != null) {
      unique_id.add(strLine1);
   }
   String unique1 = unique_id.get(0);
   String unique2 = unique_id.get(1);
    String   key1 = Xor(card1,unique1);
      System.out.println(key1);
      key1=getMD5(key1);
      key2=getMD5(key2);
      System.out.println(key1);
   statement = connection.createStatement();
   String query = "SELECT keysss FROM info";
   resultSet = statement.executeQuery(query);
   keys = new ArrayList<String>();
   while (resultSet.next()) {
      String rs = resultSet.getString("keysss");
      keys.add(rs);
   }
setBounds(250, 150, 500, 200);
setTitle("RFID card");
setLayout(new FlowLayout());
box1 = new Box(BoxLayout.X_AXIS);
box = new Box(BoxLayout.X_AXIS);
label = new JLabel("Card info");
passfield = new JTextField(32);
button = new JButton("ok");
      timer = new Timer();
      RemindTask rt = new RemindTask(timer);
      timer.scheduleAtFixedRate(rt,100,4000);

      Handler h = new Handler(timer,rt);
      button.addActionListener(h);

box1.add(label);
box1.add(passfield);
box.add(box1);
box.add(button);
add(box);
}

   private String Xor(String card1, String unique1) {
      BigInteger a=new BigInteger(card1);
      BigInteger b=new BigInteger(unique1);
      BigInteger c=a.xor(b);
      return c.toString();
   }

   @Override
   public void run() {
```

```
        while(true) {
            repaint();
            try {
                Thread.sleep(100);
            } catch (InterruptedException ex) {

Logger.getLogger(RFID.class.getName()).log(Level.SEVER
E, null, ex);
            }
        }
    }

class RemindTask extends TimerTask{
        private String need;
        Timer t;
        public RemindTask(Timer timer)
        {
            t = timer;
        }
    public void run() {
        String cut = passfield.getText();
        repaint();
        if(!cut.isEmpty()){
            if(cut.length()==64){
                qq = cut.substring(0, 32);
                need = cut.substring(32,64);
            }
            else {
            need = cut;
            }
           passfield.setText(need);
        }

        System.out.println("ura!!!");
      }

}
class Handler implements ActionListener
{
    Timer t;
    RemindTask rt;

    public Handler (Timer timer, RemindTask rtask)
    {
        t = timer;
        rt = rtask;
    }
    public void actionPerformed(ActionEvent actionEvent) {

        rt.cancel();
        // t.cancel();
          icon = new
ImageIcon("C:\\diploma\\RFID\\shaika.JPEG");

       String name = null;
        String department = null;
        String statuses=null;
        String mix = null;
        boolean ok = false;
            for (int i = 0; i < keys.size(); i++) {
              if (passfield.getText().equals(keys.get(i)) &&
qq.equals(card_id.get(i))){
                    ok = true;
                    try {
                        statement = connection.createStatement();
                        resultSet = statement.executeQuery("SELECT
name, department, statuses FROM info WHERE keysss = '" +
passfield.getText() + "'");
                    while (resultSet.next()) {
                        name = resultSet.getString("name");
                        department =
resultSet.getString("department");
                        statuses=resultSet.getString("statuses");
```

```
                mix = name + " " + department;


                JOptionPane.showMessageDialog(null,
"Name: "+name+"\nFaculty: "+department+"\nStatus:
"+statuses, "Information about student",
JOptionPane.INFORMATION_MESSAGE, icon);

              }

              } catch (Exception exc) {}

            }

          }

            if (!ok) {

              JOptionPane.showMessageDialog(null, "Attempt
of copy or not registered card", "error
message",JOptionPane.ERROR_MESSAGE);

            }
```

```
        java.util.Timer tt = new java.util.Timer(true);

        rt = new RemindTask(tt);

        t.scheduleAtFixedRate(rt,100,4000);

        }

}


public static void main(String[] args) throws Exception,
NoSuchAlgorithmException {

    frame = new RFID();

    frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}

}
```

5. REFERENCES

[1] Nurbek Saparkhojayev and Selim Guvercin, "Attendance Control System based on RFID-technology", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, May 2012.

[2] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems." Security in Pervasive Computing, 2003, pp 201-212.

[3] F. Silva, V. Filipe, A. Pereira, "Automatic control of students' attendance in classrooms using RFID", in 3rd International Conference on Systems and Networks Communication, 2008, pp 384-389.

[4] S.Kadry and M.Smaili, "Wireless attendance management system based on iris recognition", Scientific Research and Essays, Vol. 5(12), 18 June 2010, pp. 1428-1435.

[5] V.Shehu and A.Dika, "Using real time computer vision algorithms in automatic attendance management systems", Proceedings of the ITI 2010 32nd International Conference on Information Technology Interfaces, 21-24 June, 2010, Cavtat, Croatia.

[6] Xiaoyun Wang, Dengguo Feng, Xuejia Lai and Hongbo Yu, "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD", 2004.

[7] I.N. Tselepis, M.P. Bekakos, A.S. Nikitakis and E.A. Lipitakis, "MD5 Hash Algorithm Hardware Realization on a Reconfigurable FPGA Platform", HERCMA CONFERENCE "COMPUTER MATHEMATICS, PROGRAMMING AND SOFTWARE APPLICATIONS", 2007.

[8] NXP official web-site. For additional information, visit http://www.nxp.com.

[9] Stronglink Technology Co, Ltd. Official website: http://www.stronglink-rfid.com.

[10] M.Dhanalakshmi and U. Mamatha, "RFID based library management system", Proceedings of ASCNT, pp.227-234,2009.

[11] S.B.Patil and R.M.Walli, "Design and Development of fully automatic AT89C52 based low cost embedded system for rail tracking", International Journal of electronic communication and soft computing science and engineering", Vol. 1, Issue 1.