ISSN 2085-4579

Sim Card Alarm for Android Smartphone

Rikip Ginanjar, Ridho Utomo
Faculty of Computing, President University, Bekasi, Indonesia
rikipginanjar@president.ac.id

Diterima 25 April 2016
Disetujui 17 Juni 2016

***Abstract*— Since the first Android smartphones were released, many applications have been developed until today. One of them is security application. Security applications consist of several types of applications such as Antivirus, SIM Card Change Alarm, and Applocker. SIM Card Changed Alarm is a security application that has capability to monitor SIM Card change in Android smartphones. Although there are many SIM Card Change Alarm applications which are available in Google Playstore, there are many people who are not satisfied with this applications. It happens due to them only sends the SMS alert to the owner without locking the phone to prevent other people using it. The outcome of this research will have the same feature as SIM Card Change Alarm but with additional features such as locking the smartphone if the SIM Card has been changed, locking and unlocking remotely using SMS, triggering loud alarm, and calling-back to owner when the smartphone has been lost or stolen. The outcome expected for the Prifone Application is a user-friendly application which could be used by many people around the world and also is able to secure Android smartphones.**

***Index Terms*—android, sim card alarm, security aplication**

I. INTRODUCTION

Nowadays, the gadget can hardly be separated from human life, especially smartphone. With the smartphones people almost can do everything like reading, taking pictures with better camera, watching HD movies, playing games, chatting, and browsing. Therefore, the existence of smartphones has become important for the people to make their lives easier.

Android has become one of the leading smartphone operating systems in comparison to other smartphone operating systems. Many smartphone products use Android as their operating system. Open source, easy to customize, and a support of thousand application by third-party. Those are the reasons why Android becomes the most popular operating system for smartphones.

As a gadget that is often used, people usually store private data such as photos, videos, and login to several social media applications like Facebook, Line, Whatsapp, and Twitter. It is really dangerous if the people lose their smartphones or stolen by the thieves. The thief can access private data on that smartphone or using account that is registered on that smartphone for something bad. The owner can do nothing to prevent that from happening. The owner usually only tries to call to the number that is used by their missing phone and it is not working because usually the thief already changes the SIM card. Based on that, this research will create an application that has features to lock the smartphones and callback to the owner if the SIM card has been changed.

II. LITERATURE STUDY

A. Subscriber Identity Module (SIM)

SIM associates a physical card used in smarphones to a subscriber of the Mobile Network Operator. The SIM's storage also includes a unique serial number ICCID (Integrated Circuit Card Identifier) which identifies the SIM globally and unique IMSI (International Mobile Subscriber Identity). SIM card usage can be controlled with two password: PIN and PUK. PUK is used as a remedy if PIN has been entered incorrectly too many times [2].

The file system of a SIM is organized in a hierarchical tree structure, it consists of the following three types of elements: Master File, Dedicated File and Elementary File [3] and according to ETSI standards [5] the SIM card provides a possibility of storing files, the

capacity of the module makes possible to store a considerable amount of keys which are less than 1 Kb in size [6].

B. Integrated Circuit Card Identifier(ICCID)

The integrated circuit card identification is a unique numeric identifier for the SIM that can be up to 20 digits long. The ICCID can be read from the SIM without providing a PIN and can be never updated [1].

C. International Mobile Subscriber Identity(IMSI)

IMSI (International Mobile Subscriber Identity), is a unique number that is associated with all GSM (Global System for Mobile Communications) and UMTS (Universal Mobile Telecommunications System) network mobile phone users. An International Mobile Subscriber Identity is up to 15 digits long. The first three digits represent the country code, followed by the network code. The remaining digits, up to fifteen represents the unique subscriber number from within the network's customer base [4].

III. SYSTEM OVERVIEW

The outcome of this research is an application which is named as *"Prifone"*. The Prifone application is one of many security applications that develop in Android operating system. The Prifone application is intended to assist the user to prevent Android smartphone can be used by others when the smartphone has been lost or stolen, keep monitoring the SIM Card change in the Android smartphone, and assist the user to get the smartphone back if the smartphone has been lost.

Prifone application allow the user to:

- Enable or Disable the SIM Card Alarm.
- Register the SIM Card based on ICCID and IMSI number on the SIM Card.
- Register the remote number that will be used by another phone user as a remote control.
- Set the lock keyword that will be used to lock the smartphone using SMS remotely. Set the unlock keyword that will be used to unlock the smartphone using SMS remotely.
- Set the setting keyword that will be used to get the current setting of Prifone using SMS remotely.
- Set the password that will be used to access Prifone application.
- Uninstall the Prifone application from inside the application.

The security objectives of this research:

- A personalized device shall only work with a specific subset of SIMs.
- Only a user with a valid unlock code shall be able to depersonalize the device.

This application will rely on SMS keyword that sent from remote number that already set in the Prifone application. There are several ways which the user can get the result of this application.

First, Prifone will lock the smartphone, trigger the alarm and run the callback to the owner, if (a) the phone receives sms that contains lock keyword from remote number, (b) the user changes the SIM Card without disabling the SIM Card alarm first, (c) Prifone detects there is no SIM Card in the smartphone and (d) the user fails three times entering the password to access Prifone application.

Second, Prifone will unlock the smartphone and stopped the alarm if the phone received sms that contains unlock keyword from remote number. Third, Prifone will automatically send SMS contains Prifone current setting if the phone received sms that contains setting keyword from remote number. Last, the Prifone will uninstall itself if the user taps uninstall in the Prifone menu.

Whenever the Profine detect different SIM Card placed in the smartphone, the system will detect the SIM Card ICCID and IMSI after boot complete. After that, system will compare new ICCID and IMSI with ICCID and IMSI that already saved in shared preference. The result is not match then the system will send SMS alert to remote number. After that, the system will lock the phone. When the phone is locked the system will trigger the loud alarm and callback to the owner as shown in Figure 1.

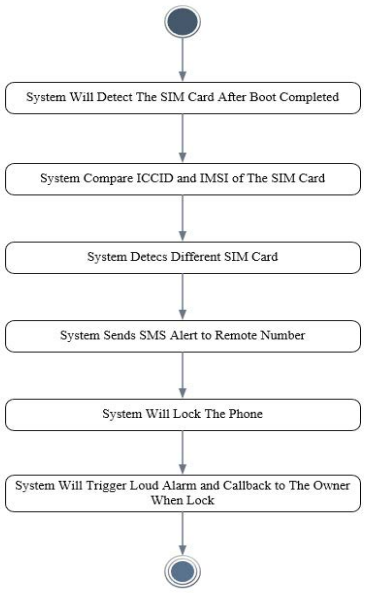

Figure 1 Activity Diagram of Unauthorized SIM

IV. SYSTEM DEVELOPMENT

A. Sdk and Permission Requirements

Figure 2 shows the sdk version and permission that used in Prifone Application. Android:minSdkVersion="11" that means the minimum android version that can run Prifone application is smartphone with operating system Android API 11 (Android Honeycomb). Prifone uses several permission there are: READ_PHONE_STATE, RECEIVE_BOOT_COMPLETED, CALL_PHONE, RECEIVE_SMS, SEND_SMS, MODIFY_AUDIO_SETTINGS and SYSTEM_ALERT_WINDOW.

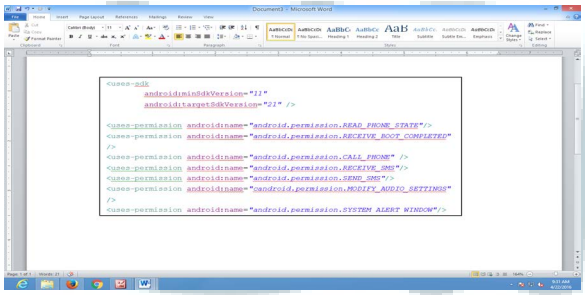

Figure 2 Sdk and Permission Requirements

Prifone need READ_PHONE_STATE permissions to detect SIM Card and read the SIM Card ICCID and IMSI. Prifone also need CALL_PHONE, RECEIVE_SMS, and SEND_SMS permission to be able receive sms keyword from remote number, send prifone current setting to remote number and automatic callback to remote number. Furthermore, prifone also need RECEIVE_BOOT_COMPLETED, MODIFY_AUDIO_SETTINGS and SYSTEM_ALERT_WINDOW permission to run on startup, change audio setting to turn on and turn of speakerphone and to draw other apps when Prifone lock screen running.

B. Register SIM Card in Prifone List Setting Menu

The first thing to do is import onpreferenceClick and implements OnPreferenceClickListener in setting activity class to listen preference click. After that in onpreferenceClick method listen register simcard preference key.

```
public boolean onPreferenceClick(Preference preference) {
        // TODO Auto-generated method stub
        String pref = preference.getKey();
```

Figure 3 OnpreferenceClick Listener Method

After application get the preference key initialize sharedpreference as place to store IMSI and ICCID SIM card, to use and edit sharedpreference system need import SharedPreferences and SharedPreferences as shown in Figure 3.

In order to detect SIM Card ICCID and IMSI the application require permission READ_PHONE_STATE that declare in application manifest file as shown in Figure 4. After that call TelephonyManager API to detect SIM Card. If application successfully detect SIM Card then it will get ICCID and IMSI SIM Card using *getSimSerialNumber()* and getSubscriberId() API and convert it to string. After get ICCID and IMSI system will edit defaultsharedpreference and put ICCID and IMSI into it then save it. Finally system will show the alert *"Sim Card Registered Successfully"* and change preference summary to *"Registration Completed"*. But, if the system detect there is no SIM Card inserted into the phone then system will show the message alert *"Sim Card Not Detected"*.

```
public boolean onPreferenceClick(Preference preference) {
            // TODO Auto-generated method stub
   String pref = preference.getKey();
     if(pref.equals(setsim.getKey())){

       SharedPreferences sim =
       PreferenceManager.getDefaultSharedPreferences(this);
       TelephonyManager simcard =
       (TelephonyManager)getSystemService(Context.TELEPHONY_SERVICE);

       if (simcard.getSimState()!= TelephonyManager.SIM_STATE_ABSENT){
           String serialnum = simcard.getSimSerialNumber().toString();
           String subscriber = simcard.getSubscriberId().toString();
           Editor editor = sim.edit();
           editor.putString("serial", serialnum);
           editor.putString("subscriber", subscriber);
           editor.commit();

           Toast.makeText(this, this.getString(R.string.simsuccess),
           Toast.LENGTH_SHORT).show();

           setsim.setSummary(Html.fromHtml(getResources().getString(R.stri
           ng.summcomplete)));

       }else{

           Toast.makeText(this, this.getString(R.string.simabsent),
           Toast.LENGTH_SHORT).show();
       }

       }else if(pref.equals(setlockkeyword.getKey())){
          String x = newValue.toString();

           if (!x.matches("")) {
             Toast.makeText(this,this.getString(R.string.lockkeywor
             dsuccess),Toast.LENGTH_SHORT).show();
             setlockkeyword.setSummary(Html.fromHtml(getResources()
             .getString(R.string.summcomplete)));
            return true;
           }else{
             Toast.makeText(this,
             this.getString(R.string.lockkeyblank),
             Toast.LENGTH_SHORT).show();
             return false;
           }
       }
```

Figure 4 Register SIM Card ICCID and IMSI

C. *Register Lock, Unlock, and Setting Keyword*

Figure 5 show code how to register lock keyword, unlock keyword and setting keyword. First, system will check which preference key that user change. If preference key equals **setlockkeyword** then system will get user input from **edittextpreference setlockkeyword**, then check the input, if the user input not equals (“”) means user input something into **edittextpreference** then it will return true means data will automatically save into **defaultsharedpreference** and the data will be stored in with the name accordance with the key name. But, if user input equals (“”) means user did not input anything in **edittextpreference** then system will show alert message *“field cannot be empty”*. The system will do the same thing with **setunlockkeyword** and **setsettkeyword**.

```
}else if(pref.equals(setunlockkeyword.getKey())){
   String x = newValue.toString();
      if (!x.matches("")) {
        Toast.makeText(this,
        this.getString(R.string.unlockkeywordsuccess),
        Toast.LENGTH_SHORT).show();

        setunlockkeyword.setSummary(Html.fromHtml(getResources().g
        etString(R.string.summcomplete)));
        return true;

      }else{

      Toast.makeText(this,
      this.getString(R.string.unlockkeyblank),
      Toast.LENGTH_SHORT).show();
      return false;
      }

}else if(pref.equals(setsettkeyword.getKey())){
    String x = newValue.toString();
      if (!x.matches("")) {
        Toast.makeText(this,
        this.getString(R.string.settkeywordsuccess),
        Toast.LENGTH_SHORT).show();

        setsettkeyword.setSummary(Html.fromHtml(getResources().get
        String(R.string.summcomplete)));
        return true;

      }else{
         Toast.makeText(this,
         this.getString(R.string.settkeyblank),
         Toast.LENGTH_SHORT).show();
         return false;
      }
```

Figure 5 Register Lock, Unlock, and Setting Keyword Code

D. *Register Password*

Figure 6 show code how to register password. First system will get the user input from password edittextpreference. After that, system will call isValidPass method to do the validation the password. Method isValidPass has function to check whether the password input contains lowercase, uppercase, and number with the length 6 – 20 character. If isValidPass return true then system automatically save the password into default shared preference. But if isValidPass return false then system will show alertdialog that tell the user if the password must contains lowercase, uppercase, number and the length 6 – 20 character.

```
private boolean isValidPass(String x) {
// TODO Auto-generated method stub
  Pattern pattern;
  Matcher matcher;

      final String PASSWORD_PATTERN =  "((?=.*\\d)(?=.*[a-
      z])(?=.*[A-Z]).(6,20)";

      pattern = Pattern.compile(PASSWORD_PATTERN);
      matcher = pattern.matcher(x);

      return matcher.matches();

}
```

Figure 6 Check Password Code

E. *Check SIM Card on Startup*

The first thing to do to make the application run on startup is uses permission RECEIVE_

BOOT_COMPLETED. After that, create Class that extends Broadcastreceiver class then filter the receiver with android.intent.action.BOOT_COMPLETED and give priority "2147483647" in android manifest file to make application run faster after boot completed as shown in Figure 7.

```
private void CheckSim(Context context) {
          // TODO Auto-generated method stub
 SharedPreferences sim =
 PreferenceManager.getDefaultSharedPreferences(context);
      String Serial = sim.getString("serial", "");
      String Subscriber = sim.getString("subscriber", "");
      Boolean enable = sim.getBoolean("enableapp", false);
      Boolean lock = sim.getBoolean("lock", false);
 SharedPreferences key =
 PreferenceManager.getDefaultSharedPreferences(context);

 TelephonyManager simcard = (TelephonyManager)
 context.getSystemService(Context.TELEPHONY_SERVICE);
      if (simcard.getSimState()!=
TelephonyManager.SIM_STATE_ABSENT){
          String Serial1 = simcard.getSimSerialNumber();
```

Figure 7 Check SIM Card on Startup

Now start to check the SIM Card, First application will get registered SIM Card ICCID and IMSI from default sharedpreference. After that, application check if there are a SIM Card inserted in smartphone. If application detect there are no SIM Card inserted then application automatically run SimAbsentActivity that lock the smartphone and ring loud alarm. But if application detect SIM Card, application will start compare IMSI and ICCID that already saved in default sharedpreference with the new one. If the IMSI and ICCID matched then application will do nothing. But, if application detect different IMSI and ICCID then application automatically run UnauthorizedSimActivity that will lock the smartphone, run the callback feature and ring loud alarm.

F. *Send Current Prifone Setting*

The first thing to do to make application able to receive SMS is add permission RECEIVE_SMS In android manifest. After that get the message body and the sender number. Now, application will compare message body with keyword, to compare it used equalsignorecase(), therefore, it will be case insensitive. To compare number it used two ways, first, it used PhoneNumberUtils. compare() it will ignore country code. For example +6285286636771 is same with 085286636771 but this method did not work on some smartphone. Therefore, prifone used the second way to compare the phone number using equals method. The Code is shown in Figure 8.

```
public void onReceive(Context context, Intent intent) {
      Bundle intentExtras = intent.getExtras();
      if (intentExtras != null) {
          Object[] sms = (Object[]) intentExtras.get("pdus");

          for (int i = 0; i < sms.length; ++i) {
             String smsMessage = SmsMessage.createFromPdu((byte[])
             sms[i]).getMessageBody().toString();

             String address = SmsMessage.createFromPdu((byte[])
             sms[i]).getOriginatingAddress();

             SharedPreferences conkey =
             PreferenceManager.getDefaultSharedPreferences(context);
            String contact = conkey.getString("setnum", "");
            String lockkey = conkey.getString("setlockkeyword", "");
            String unlockkey = conkey.getString("setunlockkeyword", "");
            String setkey = conkey.getString("setsetkeyword", "");

           if (PhoneNumberUtils.compare(contact, address) ||
(address.equals(contact))){
              if (smsMessage.equalsIgnoreCase(lockkey)){
                    Editor editor = conkey.edit();
                    editor.putBoolean("lock", true);
                    editor.putBoolean("enableapp", true);
                    editor.commit();
                    abortBroadcast();
                    Intent Absent = new Intent(context,
                    LockActivity.class);
                    Absent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                    context.startActivity(Absent);

             }else if (smsMessage.equalsIgnoreCase(unlockkey)){
                    TelephonyManager simcard = (TelephonyManager)
                    context.getSystemService(Context.TELEPHONY_SERVICE);
                    String Serial1 = simcard.getSimSerialNumber();
                    String Subscriber1 = simcard.getSubscriberId();
                    Editor editor = conkey.edit();
                    editor.putString("serial", Serial1);
                    editor.putString("subscriber", Subscriber1);
                    editor.putBoolean("lock", false);
                    editor.commit();
                    abortBroadcast();
                    Toast.makeText(context,
                    context.getString(R.string.simupdate),
```

Figure 8 SMS Receiver Lock and unlock Code

G. *Start and Stop Alarm*

Figure 9 show the code to Start and Stop Alarm, to make application able to change the volume application required MODIFY_AUDIO_SETTING permission.

```
public void StartAlarm (){
 audiomanager =
 (AudioManager)getApplicationContext().getSystemService(AUDIO_SERVICE);
 audiomanager.setStreamVolume(AudioManager.STREAM_MUSIC,
 audiomanager.getStreamMaxVolume(AudioManager.STREAM_MUSIC),0);
 audiomanager.setMode(AudioManager.MODE_IN_CALL);
 audiomanager.setSpeakerphoneOn(true);
 homebutton.lock(this);

 if (mediaplayer == null){
 mediaplayer = MediaPlayer.create(getApplicationContext(), R.raw.beep1);
 mediaplayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
 mediaplayer.setOnCompletionListener(new OnCompletionListener(){
     public void onCompletion(MediaPlayer mediaplayer){
          if (isTrue() == false){
               homebutton.unlock();

      audiomanager.setMode(AudioManager.MODE_NORMAL);
                    audiomanager.setSpeakerphoneOn(false);
                    mediaplayer.stop();
                    mediaplayer.reset();
                    mediaplayer.release();
                    mediaplayer = null;
                    finish();
          }else{
               mediaplayer.start();}
               }
          });
          mediaplayer.start();
          }
```

Figure 9 Code to Start and Stop Alarm

H. Make a Phone Call

Figure 4.9 show the code to make phone call, to make application able make phonecall application required CALL_PHONE permission and to turn on speakerphone automatically application also required MODIFY_AUDIO_SETTING.

```
Intent intent = new Intent(Intent.ACTION_CALL);
intent.setData(Uri.parse("tel:"+Num));
startActivity(intent);
```

Figure 10 Code to Make a Phone Call

V. USER INTERFACE

a. Login Screen

Login screen will show every time when user wants to access Prifone Application if user already set the password before. Login screen has function to protect Prifone that cannot be accessed by others. Therefore, only the owner that can change the Prifone setting.

b. Setting Menu Screen

Figure 11 shows the setting menu of the Prifone application. Prifone made with a simple design and easy to use by everyone. Therefore, all the settings are presented in one list. Prifone menu consists of several features that are, Enable SIM Card Alarm, Detect SIM Card ID, Set Remote Number, Set Lock Keyword, Set Unlock Keyword, Set Setting Keyword, Password, and Uninstall Prifone.

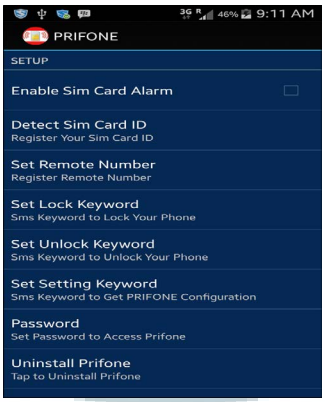

Figure 11 Setting Menu Screen

c. Remote Number

Figure 12 shows the Set Remote Number Edit textbox, Set Remote Number edit textbox will show if user taps Set Remote Number in Prifone menu. This edit textbox needed to get number that inserted by the user. The keyboard automatically set only number input. If user already set remote number then want to change the remote number, user just tap set remote number again, remove the old number and replace with the new number. After that tap OK and it will automatically replace the old number with the new number.

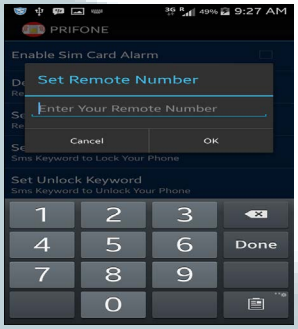

Figure 12 Set Remote Number Interface

d. Set Lock Keyword

Set Lock Keyword edit textbox will show if user taps Set Lock Keyword in Prifone menu. This edit textbox needed to get lock keyword that inserted by the user. If user already set lock keyword then want to change lock keyword, user just tap set lock keyword again, remove the old keyword and replace with the new lock keyword. After that tap OK and it will automatically replace the old lock keyword with the new lock keyword.

e. Set unlock keyword

Figure 13 shows the Set Unlock Keyword Edit textbox, Set unlock Keyword edit textbox will show if user taps Set Unlock Keyword in Prifone menu. This edit textbox needed to get unlock keyword that inserted by the user. If user already set unlock keyword then want to change unlock keyword, user just tap set unlock keyword again, remove the old keyword and replace with the new unlock keyword. After that tap OK and it will automatically replace the old unlock keyword with the new unlock keyword.

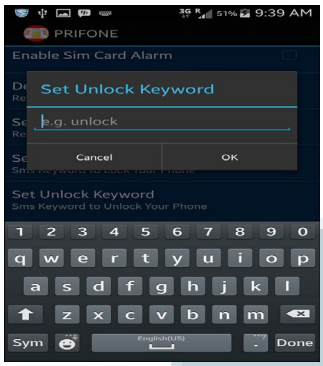

Figure 13 Set Unlock Keyword Interface

f. Lock Screen

Lock screen will show if the application receive SMS that contain lock keyword sent by remote number or user failed three times entering password in login screen. When lock screen showed, it is also run callback to the remote number and triggering loud alarm. When this happen the only way to unlock the phone and stopping the alarm is send SMS that contains unlock keyword using remote number that already set in Prifone menu.

g. Unauthorized SIM lock Screen

Unauthorized SIM Lock screen will show if the application detects different SIM Card has been inserted on start up. When Unauthorized SIM lock screen showed, it is also run callback to the remote number and triggering loud alarm. When this happen the only way to unlock the phone and stopping the alarm is send SMS that contains unlock keyword using remote number that already set in Prifone menu.

h. Absent SIM Card Lock screen

Absent SIM Lock screen will show if the application detects there are no SIM Card has been inserted on start up. When Absent SIM Card lock screen showed, it will triggering loud alarm. When this happen the only way to unlock the phone and stopping the alarm is remove the battery then insert SIM Card after that turn on the smartphone then send SMS that contains unlock keyword using remote number that already set in Prifone menu.

VI. CONCLUSIONS

This research is aim to developed one of the security applications for Android smartphones that can lock the smartphones if the SIM Card has been changed. The owner of the smartphone also can lock the smartphone remotely using SMS if the smartphone has been lost or stolen. This application also has a feature to ring loud alarm that can make the thieves become panic. Furthermore, this application also has callback to the owner feature that can help if the smartphone has been lost and someone found it, the people can tell the owner where the smartphone is. This application can run best when the smartphone has enough balance to make phone calls. However, this application still has weaknesses such as not supporting dual SIM and calling back feature that can run if the smartphone has enough balance.

REFERENCES

[1] Wayne Jansen, Rick Ayers, “Forensic Software Tools for Cell Phone Subscriber Identity Moduls”. National Insitute of Standards and Technology, 2007.

[2] Riku Itampuro, “Smartphone as Home Network’s Trusts Anchor”, Master of Science Thesis, Tempere University of Tehchnology, 2015

[3] Sheng He, “Sim Card Security”, Seminar Work, Ruhr-University of Bochum, 2007.

[4] Webopedia. IMSI Definition. Retrieved from: http://www.webopedia.com/TERM/I/IMSI.html.

[5] ETSI Technical Specification 100 977 v8.13.0

[6] Gyorgy Kalman, Josef Noll, “Sim as a key of user identification: enabling seamless user identity management in communication networks”, UNIK, 2006.