A Software Implementation of Mobile Phone Network Locking Scheme Based on Encryption Algorithm

Min Zhou

School of Intelligence Science and Information Engineering, Peihua University, Xi'an 710100, China.

Abstract. This paper introduces a mobile terminal network locking scheme based on encryption, and gives the development process of the related PC-side network locking software; this scheme is highly confidential and difficult to crack. The development of this scheme and software greatly enhances the network locking function of mobile terminal products, and provide a good commercial prospect for the mobile terminal products.

Keywords: encryption, mobile terminal, lock network.

1. Introduction

PXXXX is a mobile terminal based on the Qualcomm MSM6XXX platform, and this company prepares to use it to break into the market in Italy HX. Because HX requires the terminal to have network locking function, so must carry out network locking when the terminals are shipped. Considering the previous network locking scheme is relatively simple and easy to crack, therefore, it is a new difficult point for us to develop a network locking solution that is not easily cracked and develop its related network locking software.

2. Solution Thinking

As everyone knows, the IMEI number of mobile phone is a unique serial number to each mobile phone, we can generate a random character string for each mobile phone, bind it with the mobile phone's IMEI number, send it to the mobile phone through a certain algorithm and save it in the database. When the mobile phone needs to be unlocked, random character string corresponding to IMEI number of this mobile phone is input. After the solution is determined, then the specific algorithm is used to implement the function, therefore, after multiple evaluations, the interactive flow algorithm between the mobile phone and the PC-side network locking software and the implementation scheme for the mobile phone-side network locking function are formulated.

3. Concrete Schemes

a. Operating principle of the terminal: The IMSianternational Mobile Subscriber Identity in SIM card is used, register the described SIM card to the network, and acquire the MCC/MNC number of the described network. The MCC/MNC number of the described network is compared with the valid MCC/MNC cell-phone number stored in described terminal in advance, according to the comparison result, allow or limit described terminal to use the data card.

The pre-stored valid MCC/MNC cell-phone number in the above-mentioned terminal is composed of the first few numbers of the IMSI number in one or more SIM cards used in the terminals, the numbers can be set in advance, these Numbers are called as pre-set numbers for IMSI numbers later. The MCC/MNC number in the described network is compared with the pre-stored valid MCC/MNC cell-phone number, when the registered network's MCC/MNC number belongs to the stored valid MCC/MNC number segment, then determine the described SIM valid, and the described terminal is permitted to use the SIM. When the MCC/MNC number in the described registered network does not belong to the stored valid MCC/MNC cell-phone number, If the state which the SIM is registered to the network is normal, namely non-roaming state, then the SIM card is determind to be ilvalid, limit the terminal to use the SIM card; if the state of the SIM card registered to the network is roaming, read the IMSI number from the SIM card, the IMSI number is compared with the pre-stored valid

MCC/MNC cell-phone number, determine whether previously set number of the IMSI number belongs to the stored valid MCC/MNC cell-phone number, if it is, then determine the described SIM card to be valid, the described terminal is permitted to use the SIM card; otherwise, determine the SIM card to be invalid, the described terminal is limited to use the SIM card.

The above process of upgrading the board can be completed by the terminal alone; the purpose which operators upgrade board is to prevent the source files of the flash and other memories from being hacked by the hackers. In practical applications, hackers may also make upgrade packages and upgrade the boards privately, so achieve the purpose of distorting the above-mentioned valid MCC/MNC cell-phone number. Therefore, after the upgrading process of the board is completed, the board needs to be carried out security check.

The process of the security check is as follows: 1. Calculate a Hash value in accordance with the multiple image files loaded on the board, package the Hash value and other contents to form a digital certificate; 2. The above digital certificate is signed by private key in the a pair of pre-set public key and private key through tool software, then, the signed digital certificate is packaged into the boot image file of the board by one-click upgrade tool software; 3. The digital certificate is conducted integrity verification with the public key in the above-mentioned pre-set pair of public and private keys, obtain the above hash value in the above digital certificate, compare the Hash value with the previously saved valid total Hash value, if the two are inconsistent, confirm that the board is insecure, Start the board self-destruction scheme and make the board unusable.

b. PC-side network locking parameters are input in advance: the PC-side network locking software used on the production line writes the valid MCC and MNC of the mobile phone, the software algorithm and process are roughly as follows:

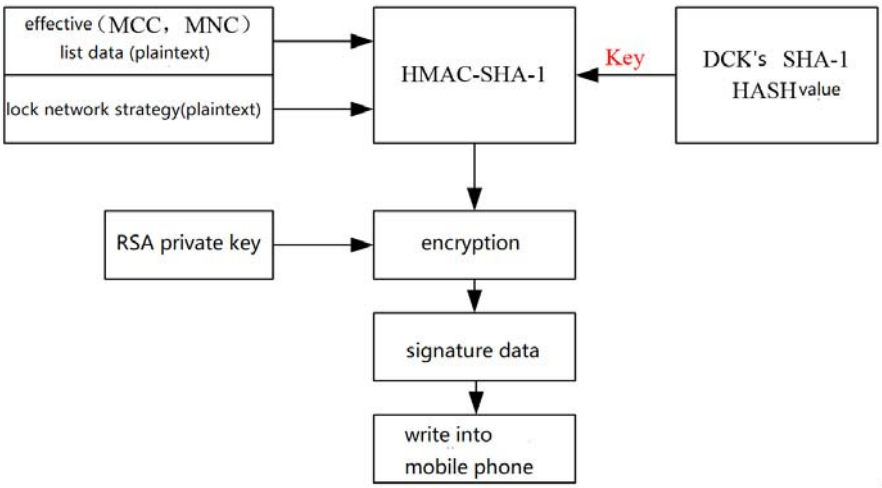

Fig. 1 Software process

Among them:

Network locking parameters (MCC, MNC): plaintext data, to be determined by the operator;

Unlock code of network locking DCK: the unlock code is randomly generated on the PC and the SHA-1 HASH value is calculated, only the SHA-1 HASH value of the DCK is saved in the mobile phone, after the network locking operation is conducted, the DCK code (plaintext) is required to save in the database along with the phone's IMEI number;

Network locking strategy: plaintext data, used to indicate whether the mobile phone needs to conduct lock status check during the starting process;

In the following, the PC-side network locking solution is mainly achieved in software:

```
SHA1_hash(unlockcode,16,DataEncrypt_PC.DCK_HASH);
  memset(key,0,20) ;
  memset(behmac,0,81) ;
  memset(outhmac,0,20) ;
  memcpy(behmac,DataEncrypt_PC.mcc_mnc_list,80);
```

```
memcpy(key,DataEncrypt_PC.DCK_HASH,20);
behmac[80]=DataEncrypt_PC.ntw_lock_policy;
hmac_sha1(behmac,81,key,20,outhmac);
// PC encrypts hmac result with private key ;
byte * From1 = outhmac ;
byte * To1 = DataEncrypt_PC.signature ;
intLength = RSA_private_encrypt(20, From1 , To1, pHandRSA, RSA_PKCS1_PADDING) ;
ofstreamout("data.dat");
if(!out)
{
    AfxMessageBox("Error open config file");
//  return FALSE;
}
/*  out <<pDst<<endl;
out.close();*/
out<<" before encryption: ";
out<<outhmac<<endl;
out<<" after encryption: ";
out<<DataEncrypt_PC.signature<<endl;
out.close();
if ( Length == 0 )
{
    if ( bVerOfChina )
        strInfo.Format("Encrypt fail!") ;
    else
        strInfo.Format("Encrypt fail!") ;
    m_ctlReportBox.InsertString(0, strInfo) ;
    returnFALSE ;
}
// send network locking command to mobile phone,
m_pPacket->bIslog = FALSE ;
m_pPacket->strLogFileName = _T("LockNetData_fromUE") ;
m_pPacket->strLogOutFileName = _T("LockNetData_ToUE" ) ;
intretnumber=m_pPacket->NewLockNet(DataEncrypt_PC.mcc_mnc_list,
DataEncrypt_PC.DCK_HASH, DataEncrypt_PC.ntw_lock_policy,DataEncrypt_PC.signature);
if ( retnumber != 0 )
{
    m_pPacket->bIslog = FALSE    ;
    if ( bVerOfChina )
    {
        if ( retnumber == 24 )
            strInfo.Format("this mobile phone network locking information has written NV,
cannot write again!") ;
        else
            strInfo.Format("fail to communicate with mobile phone during network locking!") ;
    }
    else
    {
        if ( retnumber == 24 )
            strInfo.Format("The phone can't be locked net, because it have been locked!") ;
        else
            strInfo.Format("communicate with mobile fail!") ;
```

```
        }
        m_ctlReportBox.InsertString(0, strInfo) ;
        strInfo.Format("Err Code: %d", retnumber ) ;
        m_ctlReportBox.InsertString(0, strInfo) ;
//      if(m_pConnectThread != NULL)
//              m_pConnectThread->ResumeThread();
        returnFALSE ;
  }
  m_pPacket->bIslog = FALSE ;
  // decrypt unlock code mobile phone
  byteSecCode[128] ;
  memset(SecCode,0,128) ;
  intDatalen = 0 ;
  byte * From = DataEncrypt_MOBILE.Unlocknetcode ;
  byte * To      = SecCode ;
  Datalen = RSA_private_decrypt(128,From,To,pCommunicateRSA,RSA_PKCS1_PADDING) ;
  if ( Datalen == 0 )
  {
        if ( bVerOfChina )
                strInfo.Format("fail to decrypt unlock code of mobile phone!") ;
        else
                strInfo.Format("Decrypt unlock code fail!") ;
        m_ctlReportBox.InsertString(0, strInfo) ;
//      if(m_pConnectThread != NULL)
//              m_pConnectThread->ResumeThread();
        returnFALSE ;
  }
  //
save unlock plain code of network locking
chartempcode[128];
  memset(tempcode, 0, 128) ;
//for ( int i = 0 ; i <Datalen ; i++ )
//      tempcode[i] = SecCode[i] + '0' ;
  CStringstrUnlockCode = _T("") ;
  for ( inti = 0 ; i<Datalen ; i++ )
  {
        byteMID = SecCode[i] ;
        StringToByte(FALSE) ;
        strUnlockCode += strMID ;
  }

  //if ( !SaveLockcodeDB(strImei, strUnlockCode) )

  for(i=0;i<16;i++)
  {
        if((unlockcode[i]>=0x00)&&(unlockcode[i]<=0x09))
        {
                unlockcode[i]=unlockcode[i]+0x30;
        }
        elseif((unlockcode[i]>=0x0a)&&(unlockcode[i]<=0x0f))
        {
                unlockcode[i]=unlockcode[i]+'a'-10;
```

```
        }
    }
    for ( i = 0 ; i< 16 ; i++ )
    {
        byteMID = unlockcode[i] ;
//      StringToByte(FALSE) ;
        strUnlockCode += byteMID ;
    }
    if ( !SaveLockcodeDB(strImei, strUnlockCode) )

    {
        if ( bVerOfChina )
            strInfo.Format("
fail to save unlock code of network locking!") ;
        else
            strInfo.Format("Save unlock net code fail!") ;
        m_ctlReportBox.InsertString(0, strInfo) ;
//      if(m_pConnectThread != NULL)
//          m_pConnectThread->ResumeThread();
        returnFALSE ;
    }
```

4. Conclusion

After software debugging, this scheme is achieve and finally pass test. Because this scheme flow and encryption algorithm are more complicated and difficult to crack, and the security in the communication process is high, this encryption-based mobile phone locking scheme has been used by many mobile terminal project teams for reference, this scheme can be promoted to other project groups and system terminal to achieve the network locking requirements.

References

[1]. Gao Jie, Zhang Yuesong, Bao Zhixiang. Research on network locking scheme for DCDS mobile terminal [J]. Modern Electronics Technique, 2015, 38(13):30-32.

[2]. Liu Yuanming, Huang Zhihao. Research and application of GSM locked terminal user model [J]. Telecom Engineering Technics and Standardization, 2015, 28(02): 63-67.

[3]. Qualcomm relative protocol 80-V1294-1_YD_DMSS_Serial_Data_ICD.pdf.