Mifare Card operation

Operation command code:

Operation command code:	description	
0x01	Polling card	M1 card operation
0x02	verification password	
0x03	Read data	
0x04	Write data	
0x05	Added/depreciated/stored	
0x06	Get version information	UI card operation
0x07	Read data Note: Returns 4page * 4bytes data	
0x08	Fast reading	
0x09	Read count	
0x0A	Read signature	
0x0B	Write data	
0x0C	Compatible with write data	
0x0D	verification password	
0x0E	End communication	
0x0F	Penetrate	
0x10	Batch card reading	
0x11	Batch card write	

Mifare interface basic operation flow
Polling card-->verification password-->operation-->End communication

doMifareCard(String paras, int timeout)
paras: Command code + parameter
timeout: time out

代码示例

```
private Handler mHandler = new Handler() {
      @Override
      public void handleMessage(Message msg) {
         super.handleMessage(msg);
```

```
        preDate = SystemClock.currentThreadTimeMillis();
        switch (msg.what) {
           case 3000:
              pos.doMifareCard("01",20);
              break;
           case 3001:
              String keyclass="Key A";
              pos.setBlockaddr("0C");
              pos.setKeyValue("FFFFFFFFFFFF");
              pos.doMifareCard("02"+keyclass,20);
              break;
           case 3002:
              pos.setBlockaddr("0C");
              pos.doMifareCard("03",20);
              break;
           case 3003:
              pos.doMifareCard("0E",20);
              break;
        default:
           break;
        }
     }
  };
```

Function description： getMifareStatusMsg();
called in the listen function of the mifare card operation to get the completion status of the current operation as follows:

```
@Override
public void onVerifyMifareCardResult(boolean arg0) {
  TRACE.d("onVerifyMifareCardResult(boolean arg0):" + arg0);
  long l = SystemClock.currentThreadTimeMillis();

  // TODO Auto-generated method stub
  String msg = pos.getMifareStatusMsg();
  if (arg0) {
     statusEditText.setText(" onVerifyMifareCardResult success"+"\n 时间" +
(l-preDate));

  } else {

     statusEditText.setText("onVerifyMifareCardResult fail"+msg);
  }
}
```

Error message table：

```
 "NFC_MIFARE_PARAM_ERROR";
 "NFC_MIFARE_TIMEOUT_ERROR";
 "NFC_MIFARE_CRC_ERROR";
 "NFC_MIFARE_NACK_ERROR";
 "NFC_MIFARE_POLL_ERROR";
 "NFC_MIFARE_OPERATION_ERROR";
 "NFC_ERR_ERROR";
 "NFC_ERR_BLOCK_ADDR";
 "NFC_ERR_READ_FAIL";
 "NFC_ERR_WRITE_FAIL";
```