|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 深圳市华曦达科技有限公司 | | 文 档 编 号 | 版本号 | 密级 |
| 文档编号 | Vx.x | 密级 |
| 文档名称 | SQLCipher接口使用说明文档 | | 日期 | 2017年8月18日 |

SQLCipher接口使用文档

C:\Documents and Settings\IBM\My Documents\dmc1.jpg

|  |  |  |  |
| --- | --- | --- | --- |
| **文档作者：** | **文小莉** | **日期：** | **2017.8.18** |
| **项目经理：** |  | **日期：** |  |
| **审 核：** |  | **日期：** |  |
| **批 准：** |  | **日期：** |  |

深圳市华曦达科技有限公司

文档历史发放及记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **变更（+/-）说明** | **作者** | **版本号** | **日期** | **批准** |
| 1 | 出行版本 | 文小莉 | V 1.0 | 2017-8-18 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目 录**

[文档历史发放及记录 2](#_Toc490839030)

[文档简要功能及适用范围 4](#_Toc490839031)

[1. 文档的简要功能 4](#_Toc490839032)

[2. 文档的适用范围 4](#_Toc490839033)

[1、 关于SQLCipher 5](#_Toc490839034)

[1.1 SQLCipher介绍 5](#_Toc490839035)

[2、 SQLCipher API介绍 5](#_Toc490839036)

[2.1 sqlite3\_open 5](#_Toc490839037)

[2.2 sqlite3\_key 6](#_Toc490839038)

[2.3 sqlite3\_rekey 6](#_Toc490839039)

[2.4 sqlite3\_exec 6](#_Toc490839040)

[2.4.1 常用的SQL命令 7](#_Toc490839041)

[2.4.1.1创建表 7](#_Toc490839042)

[2.4.1.2 INSERT操作 7](#_Toc490839043)

[2.4.1.3 SELECT操作 7](#_Toc490839044)

[2.4.1.4 UPDATE操作 7](#_Toc490839045)

[2.4.1.5 DELETE操作 7](#_Toc490839046)

[2.5 sqlite3\_close 8](#_Toc490839047)

[3、 SQLCipher使用实例 8](#_Toc490839048)

[3.1 加密SQLite数据库 8](#_Toc490839049)

[3.1.1 添加SQLCipher 8](#_Toc490839050)

[3.1.2 编写Android.mk文件 9](#_Toc490839051)

[3.1.3 加密SQLite数据库的实现 9](#_Toc490839052)

[3.1.3.1 创建加密数据库 9](#_Toc490839053)

[3.1.3.2 操作SQLCipher创建的数据库 10](#_Toc490839054)

[4、 SQLCipher使用补充说明 11](#_Toc490839055)

文档简要功能及适用范围

1. 文档的简要功能

文档简要介绍SQLCipher的使用说明

1）SQLCipher API操作说明

2）SQLCipher实例讲解

1. 文档的适用范围

本文档适用于SDMC研发软件工程师。

# 关于SQLCipher

## SQLCipher介绍

SQLCipher基于SQLite，因此，大多数可访问的API与SQLite 3的C / C ++接口相同。但是，SQLCipher也以PRAGMA，SQL函数和C函数的形式添加了许多安全特定的扩展。

SQLCipher是开源的，托管在github上，实现了SQLite官方的加密接口。

注意：本文使用接口为SQLCipher的c/c++接口。

# SQLCipher API介绍

## sqlite3\_open

**函数原型：sqlite3\_open(const char \*filename, sqlite3 \*\*ppDb)**

**函数功能：打开一个指向 SQLite 数据库文件的连接**

**返回：一个用于其他 SQLite 程序的数据库连接对象**

注意：如果文件名 filename 不为 NULL，那么 sqlite3\_open() 将使用这个参数值尝试打开数据库文件。如果该名称的文件不存在，sqlite3\_open() 将创建一个新的命名为该名称的数据库文件并打开。

如果 *filename* 参数是 NULL 或 ':memory:'，那么 sqlite3\_open() 将会在 RAM 中创建一个内存数据库，这只会在 session 的有效时间内持续。

## sqlite3\_key

**函数原型：sqlite3\_key(sqlite3 \*db, const void \*pkey, int nkey)**  
**函数功能：加密SQLite数据库**

**返回：SQLITE\_OK --成功 其他：失败**

**参数：第一个参数 sqlite3 是打开的数据库对象**

**pkey 是输入的密码**

**nkey 是密码长度**

该例程加密之前调用 sqlite3\_open() 打开的数据库连接。调用该接口的条件：

1. 当打开一个全新的数据库时
2. 打开一个已经加密的数据库时

## sqlite3\_rekey

**sqlite3\_rekey(sqlite3 \*db, const void \*pkey, int nkey)**

该例程对已经加密的数据库进行修改密码和清除密码的操作。调用该接口前必须使用sqlite3\_key正确输入密码。

## sqlite3\_exec

**函数原型：sqlite3\_exec(sqlite3\*, const char \*sql, sqlite\_callback, void \*data, char \*\*errmsg)**  
**函数功能：提供了一个执行 SQL 命令的快捷方式**

**返回：SQLITE\_OK—成功， 其他：zErrMsg**

**参数：第一个参数 sqlite3 是打开的数据库对象，**

**sqlite*\_*callback 是一个回调**

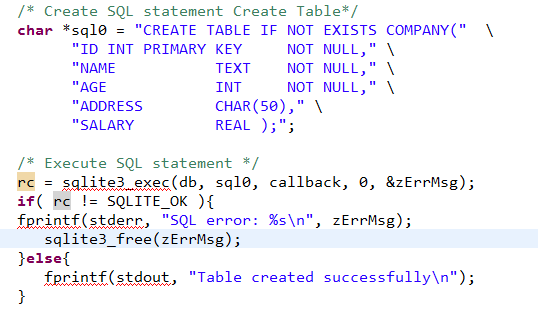
**data 作为其第一个参数**

**errmsg 将被返回用来获取程序生成的任何错误**

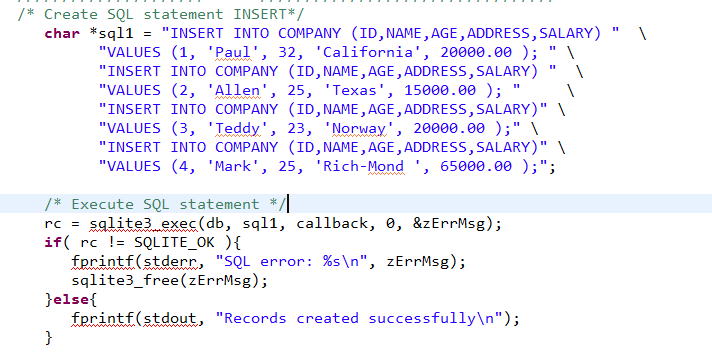
注意：sqlite3\_exec() 程序解析并执行由 **sql** 参数所给的每个命令，直到字符串结束或者遇到错误为止。

### 常用的SQL命令

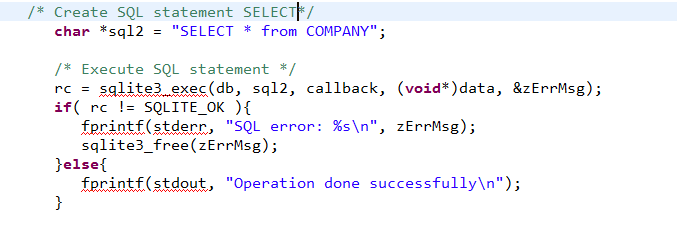
2.4.1.1创建表



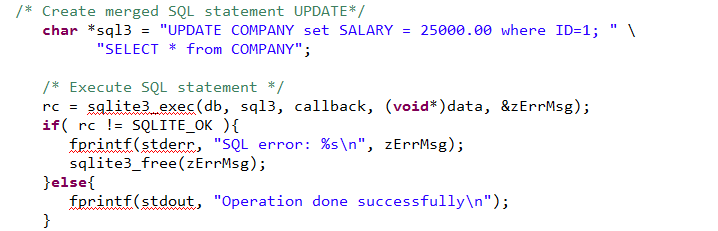
2.4.1.2 INSERT操作



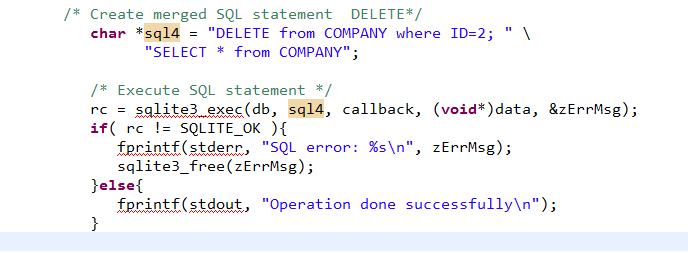
2.4.1.3 SELECT操作



2.4.1.4 UPDATE操作



2.4.1.5 DELETE操作



## sqlite3\_close

**函数原型：sqlite3\_close(sqlite3 \*db)**

**函数功能：关闭数据库连接**

**返回：如果还有查询没有完成，sqlite3\_close() 将返回 SQLITE\_BUSY 禁止关闭的错误消息。**

**参数：连接的数据库对象**

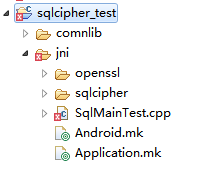
# SQLCipher使用实例

使用SQLCipher的前提是拥有SQLCipher的库文件，并且保证这个库文件是Android版的。如果没有，可以从GitHub上下一份SQLCipher的源码，编译生成sqlcipher.so文件。

## 加密SQLite数据库

### 添加SQLCipher

按如图所示，建一个工程目录，把sqlcipher.so放到comnlib下：



并把相应的openssl、sqlcipher的头文件添加到对应的目录中。

### 编写Android.mk文件

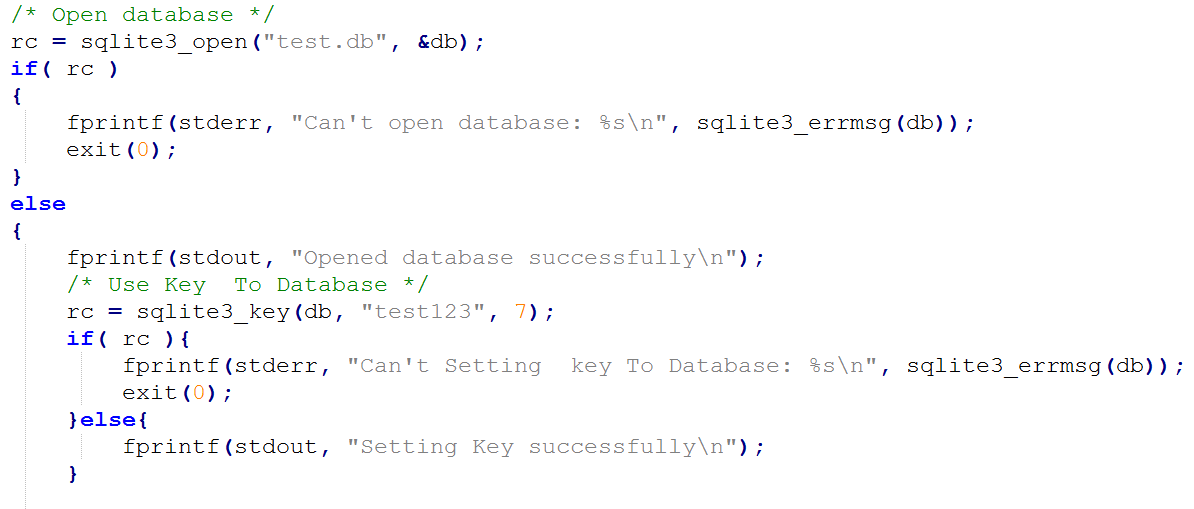
把SQLCipher的so库加载到库中调用：

LOCAL\_LDLIBS **+**= -L ./comnlib -lcrypto -lssl -lsqlcipher\_android \

### 加密SQLite数据库的实现

#### 创建加密数据库

代码如下：

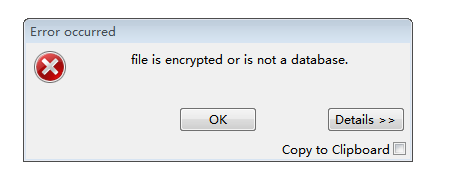


1. 在ndk下编译生成可执行文件
2. 在Android盒子里面执行:

127|root@android:# ./sqlcipher\_test

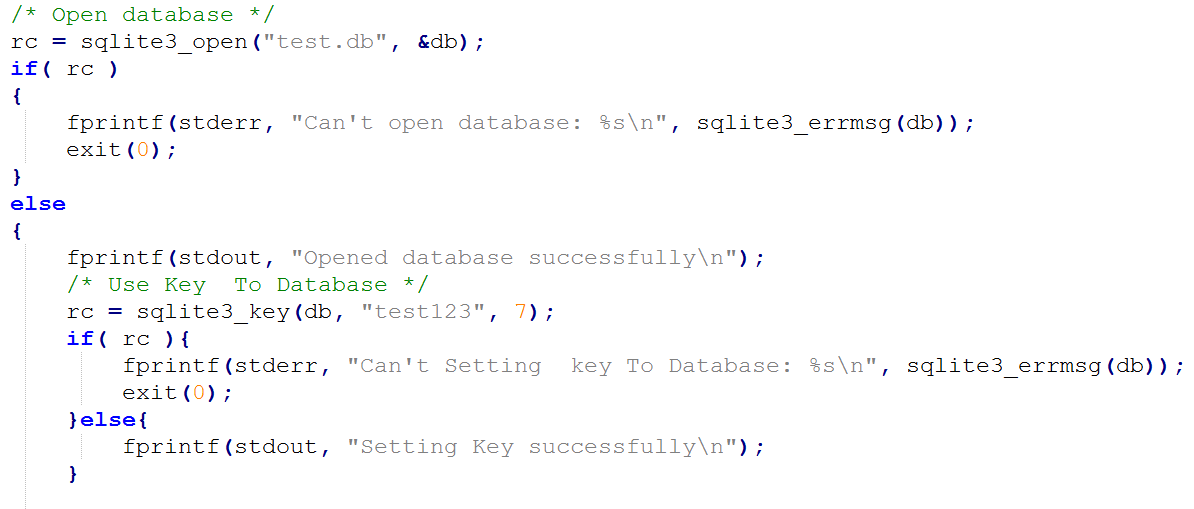
用SQLite工具打开test.db如下图所示，说明创建加密数据库成功。



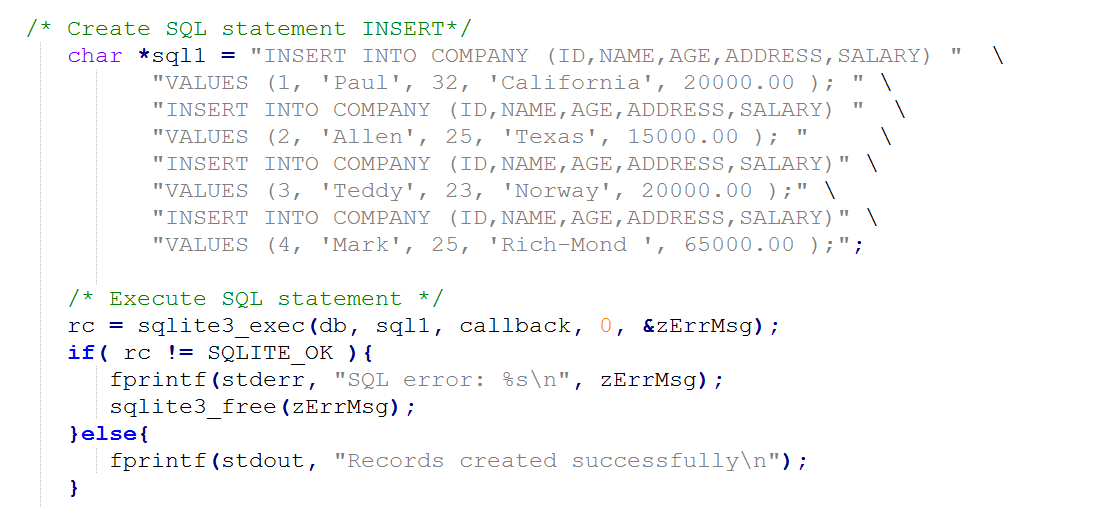
#### 操作SQLCipher创建的数据库

以INSERT和SELECT操作为例，其他类似：

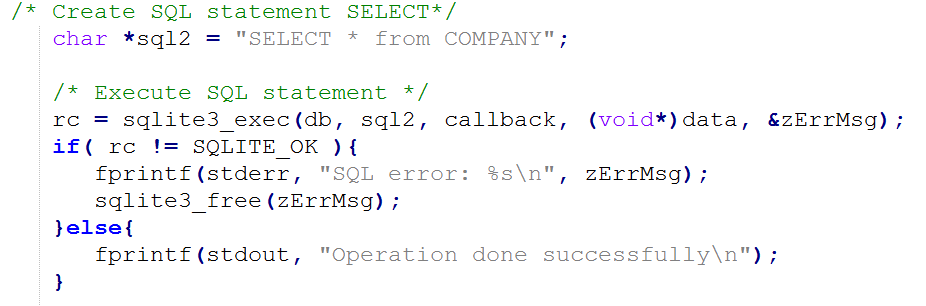
1）打开SQLCipher加密的数据库，并用sqlite3\_key输入密码：



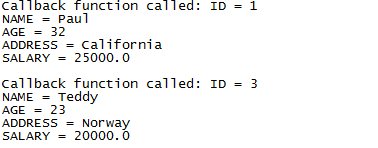
1. 用sqlite3\_exec执行“insert” sql语句



1. 用sqlite3\_exec执行“select” sql语句



1. 结果如下，在正确输入密码之后，能正常对加密数据库进行操作

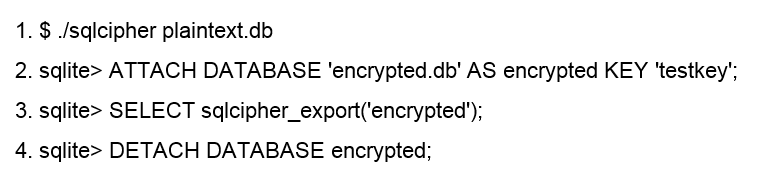


如果密码不正确或是操作加密数据库之前不使用sqlite3\_key接口输入密码，则会报出如下错误：



# SQLCipher使用补充说明

1. SQLCipher的加密接口sqlite3\_key()不能对明文数据库加密
2. 使用SQLCipher创建的数据库第三方数据库工具不能打开
3. 如果要对明文数据库加密或解密，只能使用SQLCipher提供的sqlcipher\_export()函数，操作方式：



1. 使用SQLCipher的操作流程：

