**ORMLite Integration with SQLCipher**

**Introduction:**

The document describes the feasibilities of integration of ORMLite with SQLCipher.

**SQLCipher:**

Below are the integration commands below.

**Community Edition Integration:**

compile 'net.zetetic:android-database-sqlcipher:3.5.4@aar'

**Commercial Edition Integration:**

% unzip sqlcipher-for-android-v3.5.4.zip

% mkdir -p app/libs

% cp sqlcipher-for-android-v3.5.4/sqlcipher.jar app/libs

% cp -r sqlcipher-for-android-v3.5.4/armeabi \

sqlcipher-for-android-v3.5.4/armeabi-v7a \

sqlcipher-for-android-v3.5.4/x86 app/src/main/jniLibs/

Finally, add the following line to the app/build.gradle file within the dependencies section:

compile fileTree(dir: 'libs', include: ['\*.jar'])

**ORMLite:**

Add below line to gradle file

compile **"com.j256.ormlite:ormlite-core:4.48"**compile **"com.j256.ormlite:ormlite-android:4.48"**

**Integration of ORMLite with SQLCipher:**

ORMLite is a library for SQLite, to support ORMLite to SQLCipher one has to copy files from ORMLite Library and change the imports from

*import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;*

To This   
**import** net.sqlcipher.database.SQLiteDatabase;  
**import** net.sqlcipher.database.SQLiteOpenHelper;

**And below are the classes need to modify:**

DemoOrmLiteSqliteOpenHelper.java

AndroidCompiledStatement.java

AndroidConnectionSource.java

AndroidDatabaseConnection.java

ApiCompatibility.java

ApiCompatibilityUtils.java

BasicApiCompatibility.java

OpenHelperManager.java

OrmLiteBaseActivity.java

In the constructor of DemoOrmLiteSqliteOpenHelper class add one more parameter for “password”, this will be supplied to

db = helper.getWritableDatabase(password);

In **AndroidConnectionSource.java** class.

From here, one can create a DataBaseHelper class with extending DemoOrmLiteSqliteOpenHelper and can create tables.

**How the Encryption is being performed in SQLCipher:**

1. AES-256 encryption by default, of all data.
2. Giving a passphrase at db = helper.getWritableDatabase(password) , will ensure the encryption.
3. Message authentication code (MAC) per page, to detect Tampering.
4. Based on OpenSSL libcrypto.
5. And changes the imports to

**import** net.sqlcipher.database.SQLiteDatabase;  
**import** net.sqlcipher.database.SQLiteOpenHelper

**Android performance for SQLite and SQLCipher for below operations are:**

Time Measured in MS(Milliseconds).

**SQLite on Nougat :**

|  |  |  |  |
| --- | --- | --- | --- |
|  | for 100 rows: | for 1000 rows: | for 10,000 rows: |
| Insert | 1130 | 10398 | 99915 |
| Update | 698 | 6760 | 60213 |
| Read | 19 | 52 | 279 |
|  |  |  |  |

**SQLite on Kitkat :**

|  |  |  |  |
| --- | --- | --- | --- |
|  | for 100 rows: | for 1000 rows: | for 10,000 rows: |
| Insert | 1269 | 11636 | 110258 |
| Update | 931 | 8247 | 78284 |
| Read | 21 | 95 | 1308 |
|  |  |  |  |

**SQCipher on Nougat:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | for 100 rows: | for 1000 rows: | for 10,000 rows: |
| Insert | 16945 | 185699 | 2511857 |
| Update | 1141 | 10266 | 86335 |
| Read | 21 | 289 | 400 |
|  |  |  |  |

**SQCipher on KitKat:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | for 100 rows: | for 1000 rows: | for 10,000 rows: |
| Insert | 35826 | 440727 |  |
| Update | 1418 | 13078 |  |
| Read | 380 | 499 |  |
|  |  |  |  |

For Android Nougat and OpenSSL support, please follow below link.

<https://www.zetetic.net/blog/2016/6/23/sqlcipher-android-release-n-support/>

SQLCipher Design : <https://www.zetetic.net/sqlcipher/design/>

SQLCipher Community : https://discuss.zetetic.net/t/internal-encryption-mechanism-and-library-used-in-sqlcipher/1808

Advantages:

1. Only a single native .so library is required for each platform.
2. Through Android N, SQLCipher Android supports armeabi, armeabi-v7a, and x86 platforms.
3. All internal and third-party library dependencies except for OpenSSL (the crypto provider) have been removed. This includes libbinder, libandroid\_runtime, libnativehelper, libcutils, libutils, and libicuc.
4. Less time is required to build the library from source.

**Important**: This upgrade is required for all applications wishing to support Android N while using SQLCipher for Android. Without this upgrade, current installations of SQLCipher for Android will crash on Android N

**License:**

For ORM Lite and SQLCipher are permissible and are under “BSD” license (BSD licenses are a family of permissive free software licenses, imposing minimal restrictions on the redistribution of covered software).

Please see the Copy right notice:

ORMLite :

<http://ormlite.com/javadoc/ormlite-core/doc-files/ormlite_9.html#License>

SQLCipher :

<https://www.zetetic.net/sqlcipher/license/>