

ImageLoader 加密解密共存

作者: [JakeRed](#)

原文链接: <https://ld246.com/article/1552638596513>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

ImageLoader加密解密共存

• 问题提出

Android开发中异步图片显示应用特别广泛，我们常常会用到imageloader、Glide、picasso或者其开源框架，各有好处，我们这边开发中经常使用的是imageloader，但是在最近的项目开发中，我们片的URL是加密的二进制流，出于项目的需要，我们需要对加密流进行解密。

• 一个简单的解决方法

出于快速开发的想法，想寻找一个特别简单的能够拿到图片InputStream，很快的找到了Glide的ImageDecoder回调，在这里面有一个decode方法，能够操作InputStream进行解密操作，然后倒入了Glide框架，这样一个项目中拥有了两个异步图片加载框架，怎么说也不是一个好方法。

• 干掉一个异步加载框架

上面提出了项目中因为解密加入了Glide框架，这样的做法是简单（其实imageloader也很简单），是我绝对不允许一个项目两个相同功能的框架，所以我要看看imageloader的源码，获取InputStream，进而在decode的时候把加密的图片给解密了，然后再使用中只需要给参数加上key，满足在使用中有差别，首先看一下源码，理解一下imageloader的工作流程：

```
/**
 * 这个方法是显示图片的时候调用的方法，存在ImageLoader.java里面我们看看这个传递的参数，
 * options里面传递了图片显示的一些参数，我接下来会通过option传递图片解密的密码。
 */
public void displayImage(String uri, ImageAware imageAware, DisplayImageOptions option
,
        ImageSize targetSize,
        ImageLoadingListener listener,
        ImageLoadingProgressListener progressListener) {
    checkConfiguration();
    if (imageAware == null) {
        throw new IllegalArgumentException(ERROR_WRONG_ARGUMENTS);
    }
    if (listener == null) {
        listener = defaultListener;
    }
    if (options == null) {
        options = configuration.defaultDisplayImageOptions;
    }

    if (TextUtils.isEmpty(uri)) {
        engine.cancelDisplayTaskFor(imageAware);
        listener.onLoadingStarted(uri, imageAware.getWrappedView());
        if (options.shouldShowImageForEmptyUri()) {
            imageAware.setImageDrawable(options.getImageForEmptyUri(configuration.resour
es));
        } else {
            imageAware.setImageDrawable(null);
        }
        listener.onLoadingComplete(uri, imageAware.getWrappedView(), null);
        return;
    }

    if (targetSize == null) {
        targetSize = ImageSizeUtils.defineTargetSizeForView(imageAware, configuration.getM
```

```

xImageSize());
    }
    String memoryCacheKey = MemoryCacheUtils.generateKey(uri, targetSize);
    engine.prepareDisplayTaskFor(imageAware, memoryCacheKey);

    listener.onLoadingStarted(uri, imageAware.getWrappedView());

    // 上面都是一些判空等基本操作，可以从这儿开始看。
    // 获取图片的软应用。
    Bitmap bmp = configuration.memoryCache.get(memoryCacheKey);
    // 我们看else，确定下载图片时候的流程，忽略有缓存之后的图片显示
    if (bmp != null && !bmp.isRecycled()) {
        L.d(LOG_LOAD_IMAGE_FROM_MEMORY_CACHE, memoryCacheKey);

        if (options.shouldPostProcess()) {
            ImageLoadingInfo imageLoadingInfo = new ImageLoadingInfo(uri, imageAware, tar
etSize, memoryCacheKey,
                options, listener, progressListener, engine.getLockForUri(uri));
            ProcessAndDisplayImageTask displayTask = new ProcessAndDisplayImageTask(eng
ne, bmp, imageLoadingInfo,
                defineHandler(options));
            if (options.isSyncLoading()) {
                displayTask.run();
            } else {
                engine.submit(displayTask);
            }
        } else {
            options.getDisplayer().display(bmp, imageAware, LoadedFrom.MEMORY_CACHE);
            listener.onLoadingComplete(uri, imageAware.getWrappedView(), bmp);
        }
    } else {
        if (options.shouldShowImageOnLoading()) {
            imageAware.setImageDrawable(options.getImageOnLoading(configuration.resourc
s));
        } else if (options.isResetViewBeforeLoading()) {
            imageAware.setImageDrawable(null);
        }

        ImageLoadingInfo imageLoadingInfo = new ImageLoadingInfo(uri, imageAware, targe
Size, memoryCacheKey,
            options, listener, progressListener, engine.getLockForUri(uri));
        // 真正处理图片是在这个task的异步线程里面
        LoadAndDisplayImageTask displayTask = new LoadAndDisplayImageTask(engine, ima
eLoadingInfo,
            defineHandler(options));
        if (options.isSyncLoading()) {
            displayTask.run();
        } else {
            engine.submit(displayTask);
        }
    }
}
}

```

接下来我们在看看LoadAndDisplayImageTask 这个类:

```
@Override
public void run() {
    if (waitIfPaused()) return;
    if (delayIfNeed()) return;

    ReentrantLock loadFromUriLock = imageLoadingInfo.loadFromUriLock;
    L.d(LOG_START_DISPLAY_IMAGE_TASK, memoryCacheKey);
    if (loadFromUriLock.isLocked()) {
        L.d(LOG_WAITING_FOR_IMAGE_LOADED, memoryCacheKey);
    }

    loadFromUriLock.lock();
    Bitmap bmp;
    try {
        checkTaskNotActual();
        // 再次看看内存里面有没有该图片
        bmp = configuration.memoryCache.get(memoryCacheKey);
        if (bmp == null || bmp.isRecycled()) {
            // 下载图片的真正方法
            bmp = tryLoadBitmap();
            // ..... 这个线程下面的代码和我们要做的事无关, 都是在缓存图片之类的
        }
    }
```

接下来看看tryLoadBitmap:

```
private Bitmap tryLoadBitmap() throws TaskCancelledException {
    Bitmap bitmap = null;
    try {
        File imageFile = configuration.diskCache.get(uri);
        // 如果已经有了那不是我要干的事, 所以跳过if
        if (imageFile != null && imageFile.exists() && imageFile.length() > 0) {
            L.d(LOG_LOAD_IMAGE_FROM_DISK_CACHE, memoryCacheKey);
            loadedFrom = LoadedFrom.DISC_CACHE;

            checkTaskNotActual();
            bitmap = decodeImage(Scheme.FILE.wrap(imageFile.getAbsolutePath()));
        }
        if (bitmap == null || bitmap.getWidth() <= 0 || bitmap.getHeight() <= 0) {
            L.d(LOG_LOAD_IMAGE_FROM_NETWORK, memoryCacheKey);
            loadedFrom = LoadedFrom.NETWORK;

            String imageUriForDecoding = uri;
            if (options.isCacheOnDisk() && tryCacheImageOnDisk()) {
                imageFile = configuration.diskCache.get(uri);
                if (imageFile != null) {
                    imageUriForDecoding = Scheme.FILE.wrap(imageFile.getAbsolutePath());
                }
            }

            checkTaskNotActual();
            // 找到了真正decode的代码, 当然decodeImage要干啥, 我们还得去看看
        }
    }
```

```

        bitmap = decodeImage(imageUriForDecoding);
        // ..... 下面的代码我又可以不看了
    return bitmap;
}

```

好吧，该看decodeImage：

```

private Bitmap decodeImage(String imageUri) throws IOException {
    ViewScaleType viewScaleType = imageAware.getScaleType();
    ImageDecodingInfo decodingInfo = new ImageDecodingInfo(memoryCacheKey, imageU
i, uri, targetSize, viewScaleType,
        getDownloader(), options);
    // 这里应该去看decode，按理论这个方法应该到头了。
    return decoder.decode(decodingInfo);
}

```

我们看到decode是接口的方法，我们看看谁来实现了这个接口ImageDecoder，很容易知道这个真的方法在BaseImageDecoder里面，最后看一下真正处理stream的方法：

```

@Override
public Bitmap decode(ImageDecodingInfo decodingInfo) throws IOException {
    Bitmap decodedBitmap;
    ImageFileInfo imageInfo;

    InputStream imageStream = getImageStream(decodingInfo);
    if (imageStream == null) {
        L.e(ERROR_NO_IMAGE_STREAM, decodingInfo.getImageKey());
        return null;
    }
    try {
        imageInfo = defineImageSizeAndRotation(imageStream, decodingInfo);
        imageStream = resetStream(imageStream, decodingInfo);
        Options decodingOptions = prepareDecodingOptions(imageInfo.imageSize, decoding
nfo);
        // decodedBitmap 就是要显示的图片了，所以我们在这里严处理掉加密图片。
        decodedBitmap = BitmapFactory.decodeStream(imageStream, null, decodingOptions);
        //这里贴一下修改之后的处理逻辑，把decodedBitmap = BitmapFactory.decodeSt
eam(imageStream, null, decodingOptions);改为下面的代码：
        if(decodingInfo.isEncryption()){
            //加密了，这里解密方法就不贴了，私有方法，贴出来也没用。
            decodedBitmap = decryptBitmap(imageStream, decodingOptions,decodingInfo);
        }else{
            decodedBitmap = BitmapFactory.decodeStream(imageStream, null, decodingOptio
s);
        }
        // ..... 省略部分代码
    return decodedBitmap;
}

```

源码我们已经追溯完毕，接下来就是改。

- 增加Options参数值，处理加密图片

既然知其所以然了，那就干吧，我们给Options增加一个是否加密的标志位，然后再增加一个解密密码变量，这里就是传递参数的问题了。

```
private final boolean isEncryption;  
private final String password;
```

我们在上面的源码中可以看到这个DisplayImageOptions类的参数是一直传递的，所以在最外面调用时候，给options加上这两个值吧，当然我给isEncryption默认值是false，这样对非加密的图片就是以前一样的处理方式，贴一下加密的调用方式吧，然后decode的方法我就直接放到刚刚的源码里了。

```
ImageLoader.getInstance().displayImage(mPicturePaths.get(i).getmDataSource(),  
    imageView, ImageLoaderUtils.getEncryptionOptions(password));  
  
// 这里也贴一下getEncryptionOptions，放在ImageLoaderUtils类里面  
public static DisplayImageOptions getEncryptionOptions(String key){  
    return new DisplayImageOptions.Builder()  
        .showImageOnLoading(R.mipmap.pre_setting_bg) //设置图片在下载期间显示的图片  
        .showImageForEmptyUri(R.mipmap.pre_setting_bg)//设置图片Uri为空或是错误的时  
        显示的图片  
        .showImageOnFail(R.mipmap.pre_setting_bg) //设置图片加载/解码过程中错误时候显  
        示的图片  
        .cacheInMemory(true)//设置下载的图片是否缓存在内存中  
        .cacheOnDisk(true)//设置下载的图片是否缓存在SD卡中  
        .imageScaleType(ImageScaleType.IN_SAMPLE_INT)//设置图片以如何的编码方式显示  
        .bitmapConfig(Bitmap.Config.RGB_565)//设置图片的解码类型  
        .imageScaleType(ImageScaleType.EXACTLY)  
        .setEncryption(true)  
        .setPassword(key)  
        .build();//构建完成  
}
```

• 最后

代码完全是自己写的，如果有什么不对的地方，还望批评指正，我的邮箱：redzkh@gmail.com，发记录一下，会有所作用，因为好记性不如烂笔头。