# 耐飞离线下载及加解密方案实现文档

## 离线下载HLS任务，UI层调用逻辑

1、UI层通过调用CreateHlsTask([in] url,[in] keyurl,[out] hash, [out] proxyurl)创建任务

2、UI层通过调用SetTaskDownload(hash, true,””);设置该任务为离线下载任务

3、UI层调用RunTask(hash)开始下载任务

4、下载过程中，通过回调得到该下载任务的m3u8连接是否失效(id=3 result=401)，若失效，则重新请求m3u8的url,调用UpdateHttpOrgUrl(hash, url)重新设置该任务的url,再调用RunTask(hash)继续下载该任务

5、下载过程中，通过回调得到该下载任务是否已下载完成(id=1 result=0)，否则处理其它回调错误

6、对已离线的任务进行播放(支持边下边播,注意：对于没有离线完成的任务，播放过程中若链接失效则必须调用StopPlay()停止播放，待重新下载任务后再重新播放，因为播放引擎拿到是已失效的m3u8文件)，直接通过设置播放引擎代理后，play(url)即可播放已加密的离线任务

7、UI层需保存该离线任务的相关信息url,keyurl,hash等，以备下次启动初始化网络库后，通过url进行代理播放，通过hash删除任务

## 密钥生成

1、创建离线下载任务时，根据任务是否加密，随机生成一个16字节的密钥，并将密钥保存中当前缓存任务目录下.key文件中，之后加解密视频都必须通过该密钥实现，没有密钥则播放不了视频，也无法为将来p2p共享数据！

## 加密实现：

M3u8离线下载合并文件前提 m3u8文件加上每一ts片文件大小签名

1. 通过解析m3u8文件获取文件总大小及每一ts片文件大小
2. 通过设置任务属性是否需要进行加密及加密密钥
3. 对数据以每16KB为小块进行AES加密, 加密后长度相等, 将加密后的数据写入文件（最后不足一块的数据也写入）（之前是以p2p分块大小(16KB的倍数16KB-4MB区间)加密，考虑到代理服务器每次读取16KB数据，不然还得读取分块大小数据进行解密再拷贝16KB数据，造成对多余的数据进行解密，消耗CPU）
4. 数据下载完成，得到加密视频文件

## 解密实现

1. 解析任务，获取该任务信息及加密属性密钥等
2. 读取指定位置的文件部分数据，指定位置不在16KB倍数上的，往前移到16KB倍数位置上，读取加密数据后对数据每16KB为小块进行解密，拷贝偏移位置的数据流返回给代理服务器
3. 得到原始ts流数据，进行播放