计算机类发明技术交底书提纲

### 技术交底书是代理人撰写专利申请文件的依据，其要针对某一发明创造（以下简称发明）主题，说清楚别人是怎么做的？别人做的有什么缺陷？我要做什么？我是怎么做的？我做的关键点在哪里？我做的有哪些优点？

1、本发明的名称

2、背景技术的方案

3、背景技术的缺陷

4、本发明的目的

5、本发明的实施方案

6、本发明的关键点

7、本发明的效果

8、背景技术和本发明的附图

一、本发明的名称

此部分简要揭示发明主题，需清楚说明本发明的产品名称和/或方法名称，不要含有人名、单位名、商标、代号和型号等非技术术语。

一种以劫持方式进行DRM视频解密的方法及装置

二、背景技术的方案

此部分简要说明别人是怎么做的？需先简要介绍本发明涉及的技术和/或产品的性质和用途等；接下来简要介绍1到2项与本发明最接近的背景技术，即与本发明有相同目的或相同技术手段或相同用途的产品或方法，最好不是综述。

背景技术既可以是文献中的技术，也可以是常识或现有产品，是文献时需详细说明背景技术的文献出处，如专利文献号，或期刊名称、卷号、期号和页码，或书籍名称、作者、出版者、版次和页码。介绍背景技术时，应当重点说明背景技术方案的功能和实现过程。

数字视频内容通常通过数字版权管理（DRM）技术进行保护，以防止未经授权的访问和复制。这些DRM系统通常在终端设备上运行，以解密和播放受保护的视频内容。然而，一些情况下，用户可能需要对视频内容进行合法的解密，以便进行备份、格式转换或其他合法用途。

已有的数字视频解密方法通常涉及破解DRM系统，这可能涉及法律风险和侵犯知识产权的问题。因此，有必要提供一种安全、合法且有效的方法，以劫持方式进行视频解密，以维护数字版权和用户合法权益的平衡。

1. 背景技术的缺陷

此部分简要说明别人做的有什么缺陷？需要客观说明各背景技术的方法或产品在工艺或性能上的不足。但最好能结合背景技术的技术方案，介绍为什么会存在这样的缺陷。

不便的用户体验：DRM系统通常需要用户进行授权、注册或登录，以获取解密所需的密钥或许可证。这可能会导致用户体验不佳，降低了用户观看视频内容的便利性。

限制性：DRM系统通常限制了用户对数字视频内容的使用，例如，可能限制了视频内容的播放设备、转码、备份等操作。这限制了用户对数字视频的自由使用。

漏洞和攻击：DRM系统存在被攻击和破解的风险。黑客和逆向工程师可能会试图破解DRM系统，以获取解密密钥或绕过保护措施，这可能导致版权侵权和非法分发。

不同DRM系统的不兼容性：不同内容提供商可能使用不同的DRM系统，这导致了DRM系统之间的不兼容性。用户可能需要安装多个DRM解密器，以便在不同平台上观看受保护的视频内容，这增加了用户的不便。

四、本发明的目的

### 此部分简要说明我要做什么？需要简要说明本发明要克服的缺陷，或要解决的问题，或要达到的目的，应当结合上述“背景技术的缺陷”进行说明。

本发明的目的就是提出一种方法不需要获取解密key通过劫持还原经过drm加密的视频数据。本发明在不需要获取解密key ，直接读取还原经过drm加密的视频数据，以实现对数字视频内容的合法解密。

五、本发明的方案

此部分详细说明我是怎么做的？需要详细说明本发明的总体技术方案，本部分内容是技术交底材料的核心部分，也是内容最多的部分，此部分的内容需要重点突出展现本发明的技术原理、实现过程、技术措施、技术改进点等，如果技术改进点在于系统结构（或硬件结构）的变化，则最好给出系统硬件结构图，并且结合系统结构图说明系统的工作原理和工作过程，如果技术改进点还包括数据处理过程或步骤，则最好还能给出类似软件设计中常用的时序图（流程图）和功能框图，特别对于某些设计算法改进型的技术方案，还必须给出改进后的算法计算式。

图1 是一种以劫持方式进行DRM视频解密的功能。该方案包括：

S10：**配置electron以播放netflix视频**

安装Electron：配置了Git SSH密钥，安装所需的32位版本的Electron,在入口文件打开Netflix网站的主窗口。在应用程序准备就绪后，它会创建窗口并加载Netflix网站的URL。然后对项目进行打包，进行EVS签名：您安装了castLabs EVS，注册了EV账户，并使用它来对打包后的应用程序进行签名，以确保它可以播放Netflix视频。

**S20：以hook方法获取netflix下载清单信息**

1.在主进程中，创建一个新的浏览器窗口（渲染进程窗口）并加载Netflix网站。在渲染进程窗口的DOM加载完毕后，将hook代码的内容读取并通过executeJavaScript注入到渲染进程中，从而实现对Netflix网页的挂钩。

1. 在hook代码中重写了 JSON.parse 和 JSON.stringify 函数，以拦截Netflix网页中的JSON数据解析和序列化，其中包含视频url

3.将拦截到的json数据传送到主进程并进行整理获取视频url、音频url、字幕url

**S30：将劫持dll复制到指定目录**

配置和启动 Electron 应用程序，复制将劫持dll，widevinecdm.dll复制到<userData>\\WidevineCdm\\4.10.2557.0\\\_platform\_specific\\win\_x86或<userData>\\WidevineCdm\\4.10.2557.0\\\_platform\_specific\\win\_x86中，替换被劫持的widevinecdm.dll

**S40 配置keeper进程和sheller进程加载内容解密模块（keeper进行为劫持进程）**

1.使用 LoadLibrary 函数加载 widevinecdm.dll（这个widevinecdm.dll是被劫持的dll文件）文件，这是一个重要的操作，因为它是CDM的主要库文件

2.使用 GetProcAddress 函数获取 widevinecdm.dll 中的一些函数地址，包括 VerifyCdmHost\_0、InitializeCdmModule\_4、CreateCdmInstance、DeinitializeCdmModule 和 GetCdmVersion。这些函数用于初始化和操作CDM，它们分别保存在\_VerifyCdmHost\_0、\_InitializeCdmModule\_4、\_CreateCdmInstance、\_DeinitializeCdmModule 和 \_GetCdmVersion。

3.最后通过调用 \_VerifyCdmHost\_0 函数进行CDM主机的验证

**S50 劫持目标函数并创建cdm对象**

1.创建VerifyCdmHost\_0、InitializeCdmModule\_4、CreateCdmInstance、DeinitializeCdmModule 和 GetCdmVersion 这些函数，并将它们导出,VerifyCdmHost\_0、InitializeCdmModule\_4、DeinitializeCdmModule 和 GetCdmVersion这些函数调用\_VerifyCdmHost\_0、\_InitializeCdmModule\_4、\_DeinitializeCdmModule 和 \_GetCdmVersion就行了

2.CreateCdmInstance函数：该函数用于创建CDM实例，接受接口版本号（interface\_version）、密钥系统信息（key\_system）、密钥系统信息长度（key\_system\_len）、host\_function等参数。它首先记录创建日志信息，然后调用\_CreateCdmInstance函数创建CDM实例。如果未成功创建实例，将返回nullptr，否则，会检查接口版本号是否为10，如果是，则创建一个MyContentDecryptionModuleProxy对象作为CDM实例的代理对象，最后返回代理对象

3.我们使用自定义的HostFunction传入\_CreateCdmInstance，在HostFunction中，该函数用于创建CDMHost对象，其中包含对CDM的控制和管理。调用host\_function，获取CDMHost的原始对象。接着，创建一个自定义cdmHost对象作为CDMHost的原始对象的代理对象，返回CDMHost对象。

**S60 配置解密环境**

配置解密环境是通过kepper进程(劫持进程)和sheller进程进行交互完成的

Keeper进程

1. SetServerCertificate：当浏览器调用此函数时，Keeper进程会将server\_certificate\_data转换为Base64并保存起来。
2. CreateSessionAndGenerateRequest：同样，当浏览器调用此函数时，Keeper进程会将init\_data（通常为PSSH数据）转换为Base64并保存。
3. OnSessionMessage：当浏览器调用此函数时，Keeper进程会将server\_certificate\_data和pssh数据发送给sheller进程，请求生成licenseRequest。然后，它将等待sheller进程生成的license并将其发送给浏览器。
4. UpdateSession：当浏览器调用此函数并提供license时，Keeper进程将license转换为Base64并将其发送给sheller进程。

Sheller进程

1. 接收到`keeper`发来的`cert`和`pssh`后，调用`CreateCdmInstance`创建`MyContentDecryptionModuleProxy\*`对象，调用`Initialize`初始化，调用`SetServerCertificate`设置`cert`,调用`CreateSessionAndGenerateRequest`设置`pssh`，并生成`liecenseRequest`  
   2.sheller进程在`OnSessionMessage`中将生成的`liecenseRequest`发送给`sheller`，并等待`sheller`进程发送`license`过来

3.拿到`license`后调用UpdateSession设置license，此时解密环境已经配置完成

**S70 初始化解码器**

下载完解密视频后，在sheller进程中，根据视频的编码类型、质量、分辨率和加密协议（可使用bento4获取，这是一个MP4分析库），调用MyContentDecryptionModuleProxy::InitializeVideoDecoder正确初始化视频解码器的配置，以确保在进行解密之前，视频解码器已准备好接受和解码受保护的视频内容。成功初始化后，函数返回0表示成功。

**S80 解密与解码**

使用MyContentDecryptionModuleProxy::DecryptAndDecodeFrame(const InputBuffer\_2\* encrypted\_buffer, VideoFrame\* video\_frame)进行解密与解码，下面是它的参数分析

InputBuffer\_2 结构体：

const uint8\_t\* data 和 uint32\_t data\_size：这些字段表示视频帧的输入数据，包括数据指针和数据大小。这是要解密和解码的视频帧的原始数据。

EncryptionScheme encryption\_scheme：指示视频帧的加密协议，可以是 kUnencrypted（无加密）、kCenc（AES-CTR 加密）或 kCbcs（AES-CBC 加密）。

const uint8\_t\* key\_id 和 uint32\_t key\_id\_size：用于标识解密密钥的 key ID 和其大小。这是用于解密受保护视频帧的密钥标识。

const uint8\_t\* iv 和 uint32\_t iv\_size：初始化向量（IV）以及其大小。IV 在某些加密模式下用于解密视频帧。

const struct SubsampleEntry\* subsamples 和 uint32\_t num\_subsamples：这些字段用于指定视频帧中不同子样本的字节数和加密字节数。通常在样本级别进行加密，这些字段描述了每个子样本的加密和未加密部分。

Pattern pattern：这是一个结构，包含有关如何对待视频帧中的字节块的信息，包括加密块和跳过块。

int64\_t timestamp：表示视频帧的显示时间戳，以微秒为单位。这有助于将视频帧按正确的顺序显示

上面这些数据都可以通过bento4获取

video\_frame是用来接受解码后的数据的

**S90 将解密后的数据重新编码**

1.创建视频流：使用avformat\_new\_stream函数创建视频流（videoStream）。

2.打开编码器：这一部分负责打开视频编码器并设置其参数。在代码中，通过avcodec\_find\_encoder找到H.264编码器，然后为编码器分配和设置了一个AVCodecContext。建议检查avcodec\_find\_encoder和avcodec\_alloc\_context3的返回值，以确保编码器被正确初始化。此外，还可以考虑设置其他编码参数，如比特率（bitrate）和帧率（framerate），以满足需求。

3.写入文件头部：在这一步，使用avformat\_write\_header函数将文件头部信息写入输出文件。这一步是必需的，但应该检查函数的返回值，以确保文件头部正确写入。

4.解码帧并进行格式转换：从解码器接收解码后的视频帧（video\_frame），然后将其进行格式转换。这里实现了两种格式的转换，YUV420P和YUV420P10LE。通过bento4获取视频中的裸流编码是YUV420P还是YUV420P10LE，然后将YUV420P10LE转换为YUV420P，因为H.264只支持YUV420P

5.编码并写入输出流：这一步是将转换后的帧数据送入编码器，编码后的数据存储在AVPacket中。通过avcodec\_send\_frame将帧送入编码器，然后通过avcodec\_receive\_packet获取编码后的数据，并将其写入输出文件。

6.写入文件尾部：最后，使用av\_write\_trailer函数写入文件尾部信息，并释放相关的上下文和内存。

至此视频解密已完成

六、本发明的关键点

此部分的作用是让专利代理人能更清楚地认识本发明技术方案的发明点或“闪光点”，便于专利代理人在撰写专利申请文件时把握正确的方向。

本发明在通过**配置electron**，**以hook方法获取netflix下载清单信息**，**劫持目标函数并创建cdm对象，并 配置解密环境，初始化解码器，进行解密与解码、将解密后的数据重新编码 用来对DRM视频进行解密。**

七、本发明的效果

### 此部分详细说明我做的有哪些优点？需要详细阐述本发明所达到的效果和优点。可以用产量、品质、收率产率、质量、精度、效率提高，能耗、原材料、工序的节省，加工、操作、控制、使用的简便，有利于环保、降低劳动强度，出现有用性能等的数据说明。最好与背景技术比较，用实验数据来说明发明效果；也可以从理论推导或特点分析来说明发明效果。当采用实验数据时，应给出必要的试验条件和方法。

### 计算机领域发明应说明其带来的突出效果，例如数据安全、资源节省等。例如，计算产品发明应说明与现有产品相比所具有的优点，如在数据处理速度、准确度、制造成本等方面的优点。

采用本发明所述，通过劫持还原经过DRM加密的视频数据，不需要解密key，就能直接将DRM加密的视频进行解密与解码，用来提供一种安全、合法且有效的方法，以劫持方式进行视频解密，以维护数字版权和用户合法权益的平衡。

八、背景技术和本发明的附图

此部分是上述各部分的辅助部分，需要清晰给出背景技术和本发明的工艺流程图和产品结构图等及其说明。此部分可有可无，可以是一幅，也可以是多幅。工艺流程图可以在框图内说明各工艺步骤名称，产品结构图可以将各部件用引出线通过阿拉伯数字统一标号。剖面图或局部放大图要单独给出图号。附图说明需简略说明各附图的名称以及各标记的含义。

计算机技术领域的附图主要包括系统结构图、数据流程图和功能框图等。

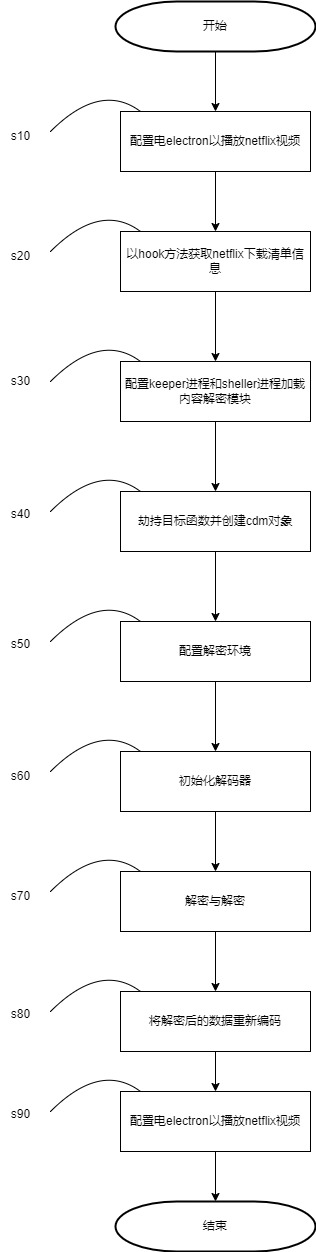


图1