

## 二、对HEVC/H.265视频编解码器进行隐写的基本思路

原创

杨酬勤 于 2021-03-27 15:33:33 发布 1281 收藏 2

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 版权协议，转载请附上原文出处链接和本声明。

本文链接：[https://blog.csdn.net/qq\\_39033409/article/details/115265319](https://blog.csdn.net/qq_39033409/article/details/115265319)

版权

### 二、对HEVC/H.265视频编解码器进行隐写的基本思路

#### 概述

- 1、视频隐写的基本思路
- 2、视频隐写的举例说明
- 3、结尾

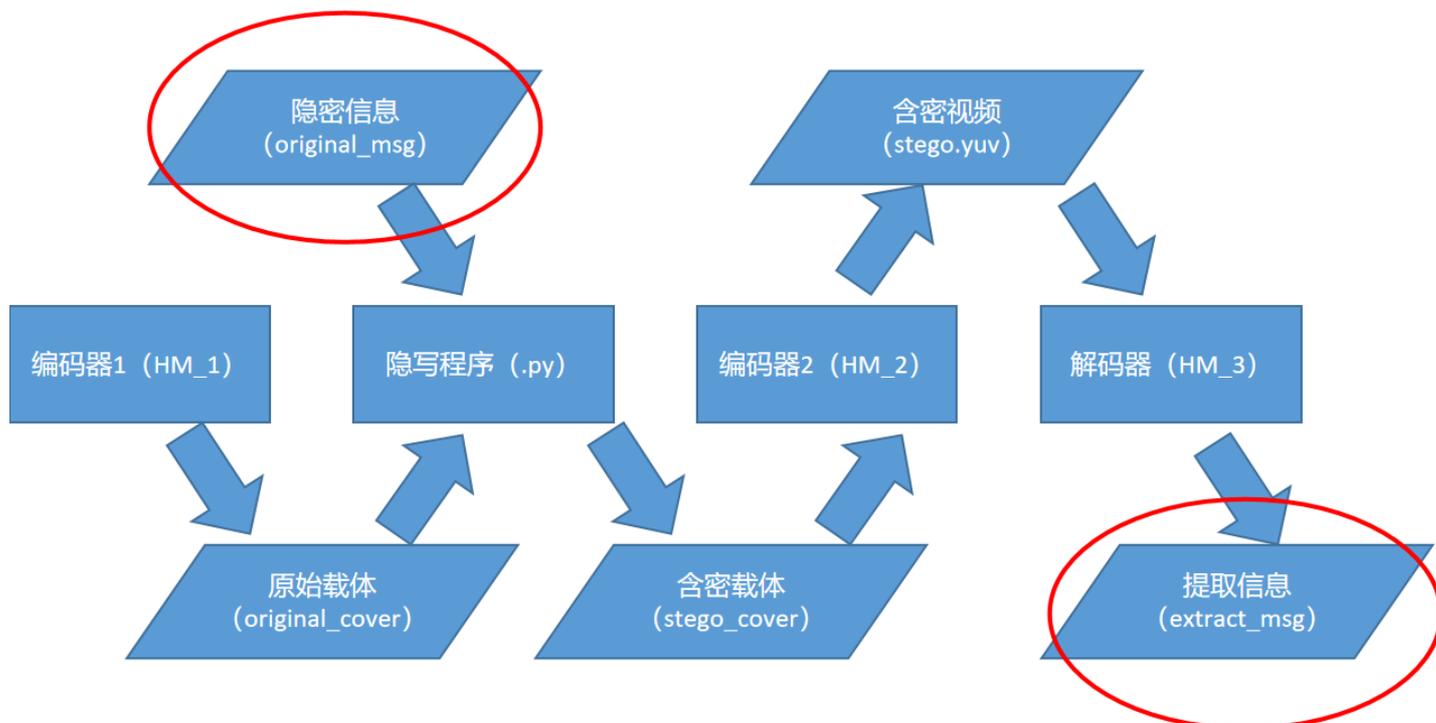
#### 概述

其实对视频隐写、图像隐写或是音频隐写，基本的思路都是一样的：读取原始图像（音频）->选择要隐写的载体->使用隐写方法对其进行隐写->产生隐写后的图像（音频）->提取隐密信息->与嵌入信息进行比对。

#### 1、视频隐写的基本思路

利用HEVC视频编解码器的隐写方法与上述过程基本相似，主要分为四个部分进行：

- 1、使用编码器1，在编码器1中提取出经选择后的原始载体信息，并将其导出保存（original\_cover）
- 2、在外部写一个隐写程序（比如用python或C++），将original\_cover的信息导入程序中，按隐写过程嵌入隐密信息（original\_msg），得到隐写完成后的含密载体（stego\_cover）
- 3、将含密载体信息传回编码器2，利用编码器2对视频进行“含密编码”，得到含密视频（stego.yuv）
- 4、将含密视频放入解码器中，提取出隐密信息（extract\_msg），并与之前嵌入的隐密信息（original\_msg）进行比对，验证嵌入和提取的信息是否一致。



[https://blog.csdn.net/qq\\_39033409](https://blog.csdn.net/qq_39033409)

## 2、视频隐写的举例说明

举一个例子，用LSB方法对帧内预测过程的帧内预测模式进行隐写：

- 1、在HM\_1的相关函数中，提取出所有的帧内预测模式，将其导出，保存成一个original\_IPM.txt
- 2、在外部的隐写程序 (.py) 中，导入original\_IPM.txt文件，并将其转换成二进制流的文件 (original\_IPM\_binary.txt)，按LSB隐写的方法，将original\_msg.txt的信息嵌入进original\_IPM\_binary.txt中，得到含密载体 (stego\_IPM.txt)

名称

-  bitstream.cfg
-  BQSquare.cfg
-  BQSquare\_416x240\_60.yuv
-  encoder\_intra\_main.cfg
-  original\_IPM.txt
-  original\_IPM\_binary.txt
-  original\_IPM\_binary\_verification.txt
-  original\_msg.txt
-  output.bin
-  output.yuv
-  stego\_IPM.txt
-  -----txt文件说明! -----.txt

[https://blog.csdn.net/qq\\_39033409](https://blog.csdn.net/qq_39033409)

- 3、在HM\_2的相关函数中，回传所有的隐写后的帧内预测模式，并按其进行编码，得到含密编码视频
- 4、将含密视频放入解码器HM\_3中，在相应的函数处，保存导出所有的帧内预测模式（stego\_IPM.txt），并将其转换成二进制流的形式（stego\_IPM\_binary.txt），按LSB的提取方法提取隐密信息（extract\_msg.txt），将提取的隐密信息与嵌入的隐密信息进行比对

### 3、结尾

这种四步骤的视频隐写方法，从理论上来说是通用的。

与直接在编码器中修改载体信息相比，这种方法虽然过程比较繁琐，且需要找到有效信息的函数位置，但是更具有普适性，且在外部隐写程序中可以调用现成的隐写方法（例如STC工具库）。

将在下一篇博客中详细记录用LSB方法对帧内预测模式进行隐写的全过程，包括所有的信息保存、隐写过程、回传过程以及在实验中遇到的问题等。