实验四、视频流 Hash

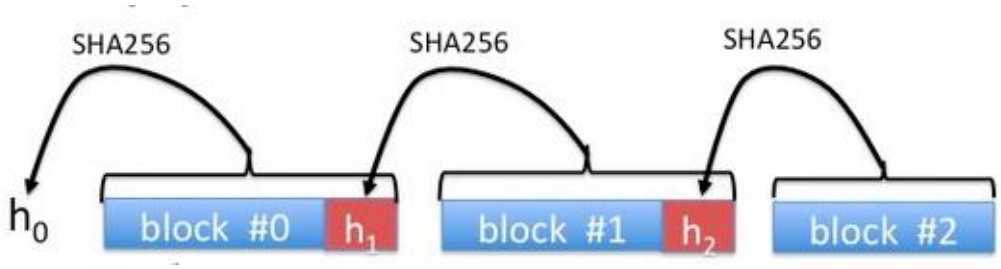
实验目的：

练习抗碰撞哈希函数的使用，理解哈希函数抗碰撞的安全性。

实验要求：

假设在一个视频网站中，浏览器下载服务器中的视频文件后需要进行认证通过后才向用户播放视频。一种直接的方法是将文件F的哈希值$h = H(F)$通过认证信道传输给用户。当浏览器将文件F全部下载完成后，可以检查 H(F)与认证信道获取的h是否相等，相等则播放视频，否则不播放。然而，使用上述方法，视频文件必须全部下载完成后才能开始播放。

为了避免上述问题，一种解决方案是网站不计算整个文件的哈希值，而是将文件分成 1KB 的数据块（1024 字节）。它计算最后一个块的哈希并将值附加到倒数第二个块。然后它计算这个增强的倒数第二个块的散列，并将得到的散列附加到最后的第三个块。此过程从最后一个块继续到第一个块，如下图所示：



网站将最终的哈希值h_0通过认证信道分发给用户。

在用户端，浏览器每次下载文件F的一个数据块，以及上述所示的附加哈希值，并进行验证，若验证通过则播放第一个视频块。具体的，通过验证$H(B_0||h_1)$与h_0是否相等来验证第一个区块的完整性； 通过验证$H(B_1||h_2)$与h_1是否相等来验证第二个区块的完整性；以此类推直至最后一个区块。这样，每

个块都会在收到时进行验证和播放，无需等到整个文件下载完毕。容易证明，只要哈希函数满足抗碰撞性质，那么攻击者的篡改行为无法通过上述完整性验证。

本次实验使用 SHA-256 作为哈希函数，实现上述视频验证程序。每一个哈希值以二进制数据的形式和视频数据块进行链接。若视频数据大小不是 1KB 的整倍数，那么最后一个区块可以小于 1KB，其余的数据区块则需要刚好是 1KB 的整倍数。

程序能够根据输入的文件 F 计算得到相应的h_0，以验证收到文件的正确性。具体的，请在实验报告中给出视频 1 的h_0值（使用 hex 编码）。

测试用例：视频 2 的h_0值为：

```
03c08f4ee0b576fe319338139c045c89c3e8e9409633bea29442e21425006ea8
```

实验报告要求:

1. 对于 SHA-256 的实现，允许使用现有的加密库，例如 PyCrypto (Python)、Crypto++ (C++) 或任何其他库。
2. 提供正确运行的程序，加上必要的注释及运行结果截图。
3. 设计、开发中的问题及实验体会。