

# base64隐写

原创

爱喝旺仔的旺旺 于 2022-01-26 09:36:54 发布 33 收藏

文章标签: [经验分享](#)

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

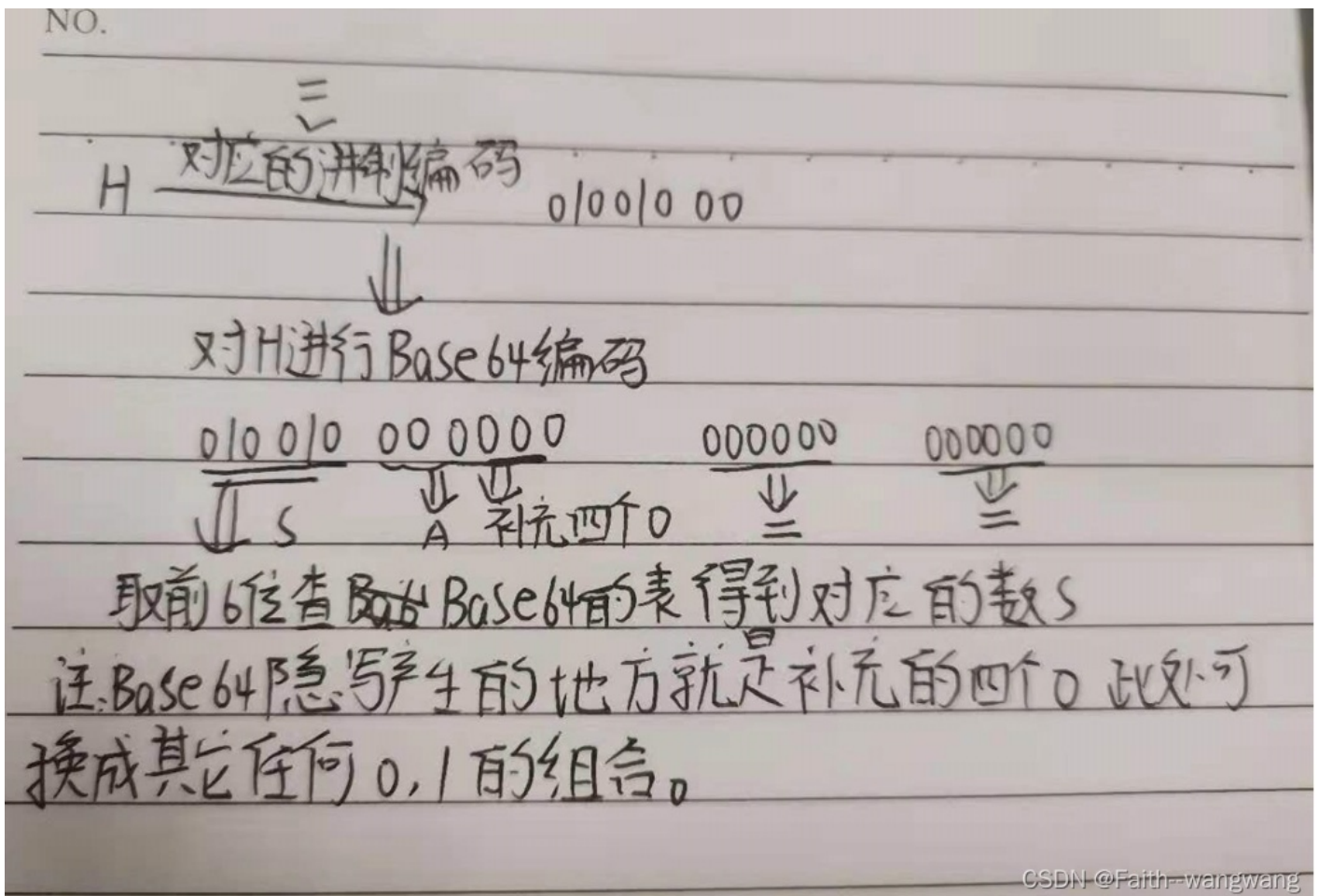
本文链接: [https://blog.csdn.net/qq\\_51233573/article/details/122695768](https://blog.csdn.net/qq_51233573/article/details/122695768)

版权

## 原理

首先了解 base64加密解密原理: 在计算机中, 采用ascii码进行编码, 转换成base64的时候先要将ascii码转化为二进制数, 在按照base64的规则进行编码。base64采用6位二进制进行编码, 不足的部分补充0; 字符是用8位二进制进行表示 (ascii码), 当对字符进行base64编码的时候。起本质就是将8位二进制转化为base64的6位二进制编码, 6, 8的最小公约数为24. 则三个字符需要用4个base64编码来表示。当不足三个字符的时候也是用四个字符来表示但是需要对字符进行特殊处理。

例如: 字符h 其二进制的编码为01001000 进行base64加密的时候只取前6位010010 查表得到对应的数S 还剩两个二进制再添四个0得到000000 查表得到A再添两个等号 == 得到。h对应的base64编码为SA==





```

import base64
b64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/ ' #导入base64能表示的所有值
with open('E:/ctf/xctf/杂项/7.zip伪加密+base64隐写/a2eb7ceaf5ab49f7acb33de2e7eed74a (1)/stego.txt', 'rb') as
    bin_str = ""
    flag = ""
    for line in f.readlines():
        stegb64 = str(line, "utf-8").strip("\n")
        # 解码再编码后, 得到隐写前的 Base64编码
        rowb64 = str(base64.b64encode(base64.b64decode(stegb64)), "utf-8").strip("\n")
        # 两次编码的末尾的数据会存在差异, 主要是由于 Base64 隐写将一部分二进制码 存到了 在编码过程中会被丢弃的二进制码中
        # 我们要做的就是将被舍弃的那一部分的二进制码找出来并拼接, 转换, 找出二进制码所表示的字符串
        # 查找每一种编码的最后一个字符的所在位置的差异 => 将会得到 相差的位数, 将位数转化成二进制码即可得到被隐写的那一部分
        offset = abs(b64chars.index(stegb64.replace('=', ''))[-1]) - b64chars.index(rowb64.replace('=', ''))[-1]
        # 每一行的 '=' 的个数 => 得到补充 0 的位数
        # 一个 '=' 表示 补了 两个二进制位
        # 两个 '=' 表示 补了 四个二进制位
        # 这些二进制位可能存在被隐写的部分
        equalnum = stegb64.count('=')
        # 包含 '=' 时 即 可能包含 Base64 隐写时
        if equalnum:
            # 将差异位 转换为 二进制 => 得到相应的二进制码并进行拼接
            # equalnum * 2 的意思为: 添加的二进制位
            bin_str += bin(offset)[2:].zfill(equalnum * 2)
    # 将二进制文件转换为字符串
    for i in range(0, len(bin_str), 8):
        flag += chr(int(bin_str[i:i + 8], 2))
    print(flag)

```