

攻防世界 crypto 入门题之转轮机加密

原创

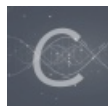
[沐一·林](#) 于 2021-08-12 11:39:32 发布 86 收藏 1

分类专栏: [CTF 密码学](#) 文章标签: [unctf](#)

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/xiao__1bai/article/details/119640312

版权



[CTF 同时被 2 个专栏收录](#)

167 篇文章 6 订阅

订阅专栏



[密码学](#)

51 篇文章 1 订阅

订阅专栏

攻防世界 crypto 入门题之转轮机加密

继续开启全栈梦想之逆向之旅~

这题是攻防世界crypto 入门题之转轮机加密

转轮机加密

👍 64 最佳Writeup由Viking • ZERO_Nu1L提供

WP

建议

难度系数: ★★★★★ 4.0

题目来源: ISCC2017

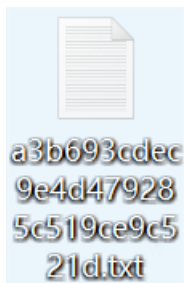
题目描述: 你俩继续往前走, 来到了前面的下一个关卡, 这个铺面墙上写了好多奇奇怪怪的 英文字母, 排列的的整整齐齐, 店面前面还有一个大大的类似于土耳其旋转烤肉的架子, 上面一圈圈的 也刻着很多英文字母, 你是一个小历史迷, 对于二战时候的历史刚好特别熟悉, 一拍大腿: “嗨呀! 我知道 是什么东西了! ”。提示: 托马斯·杰斐逊。 flag, 是字符串, 小写。

题目场景: 暂无

题目附件: 附件1

https://blog.csdn.net/xiao__1bai

下载附件:



好了, 记住了, 以后这个内容格式的就是转轮机加密了:

- 1: < ZWAXJGDLUBVIQHKYPNTCRMOSFE <
- 2: < KPBELNACZDTRXMJQOYHGVSFUWI <
- 3: < BDMAIZVRNSJUWFHTEQGYXPLOCK <
- 4: < RPLNDVHGFCUKTEBSXQYIZMJWAO <
- 5: < IHFRLABEUOTSGJVDKCPMNZQWXY <
- 6: < AMKGHIWPNYCBFZDRUSLOQXVET <
- 7: < GWTHSPYBXIZULVKMRAFDCEONJQ <
- 8: < NOZUTWDCVRJLXKISEFAPMYGHBQ <
- 9: < XPLTDSRFHENYVUBMCQWAOIKZGJ <
- 10: < UDNAJFBOWTGVRSCZQKELMXYIHP <
- 11: < MNBVCXZQWERTPOIUYSKDJFHG <
- 12: < LVNMCXZPQOWEIURYTASBKJDFHG <
- 13: < JZQAWSXCDEFVBGTYHNUMKILOP <

密钥为: 2,3,7,5,13,12,9,1,8,10,4,11,6

密文为: NFQKSEVOQOFNP

https://blog.csdn.net/xiao__1bai

原理就是转齿轮把一个字母换成另一个, 直接上一个修改后的大佬脚本:

```

rotor = [ #这里是要输入的转轮机原始字符串
    "ZWAXJGDLUBVIQHKYPNTCRMOSFE", "KPBELNACZDTRXMJQOYHGVSFUWI",
    "BDMAIZVRNSJUWFHTEQGYXPLOCK", "RPLNDVHGFCUKTEBSXQYIZMJWAO",
    "IHFRLABEUOTSGJVDKPMNZQWXY", "AMKGHIWPNYCJBFZDRUSLOQXVET",
    "GWTHTSPYBXIZULVKMRAFDCOEONJQ", "NOZUTWDCVRJLXKISEFAPMYGHBQ",
    "XPLTDSRFHENYVUBMCQWAOIKZGJ", "UDNAJFBOWTGVRSCZQKELMXIHP",
    "MNBVCXZQWERTPOIUVALSKDJFHG", "LVNCMXZPQOWEIURYTASBKJDFHG",
    "JZQAWSXCDERFVBGTYHNUMKILOP"
]

cipher = "NFQKSEVOQOFNP" #这里是要输入转轮机密文

key = [2,3,7,5,13,12,9,1,8,10,4,11,6] #这里是要输入转轮机密钥

tmp_list=[]

for i in range(0, len(rotor)):
    tmp=""
    k = key[i] - 1
    for j in range(0, len(rotor[k])):
        if cipher[i] == rotor[k][j]:
            if j == 0:
                tmp=rotor[k]
                break
            else:
                tmp=rotor[k][j:] + rotor[k][0:j]
                break
    tmp_list.append(tmp)
# print(tmp_list)

message_list = []
for i in range(0, len(tmp_list[i])):
    tmp = ""
    for j in range(0, len(tmp_list)):
        tmp += tmp_list[j][i]
    message_list.append(tmp)

print(message_list)

def spread_list(lst):
    for item in lst:
        if isinstance(item,(list,tuple)):
            yield from spread_list(item)
        else:
            yield item
        pass

if __name__ == '__main__':
    for i in spread_list(message_list):
        print("***25)
        print(i) #在多个输出中查找有语义的字符串即为flag内容

```

解毕，敬礼。