re

a_cup_of_tea

tea算法

```
from ctypes import *


def decrypt(v, k):
    v0, v1 = c_uint32(v[0]), c_uint32(v[1])
    delta = 0x543210DD
    k0, k1, k2, k3 = k[0], k[1], k[2], k[3]

    total = c_uint32(delta * -32)
    for i in range(32):
        v1.value -= ((v0.value << 4) + k2) ^ (v0.value + total.value) ^
((v0.value >> 5) + k3)
        v0.value -= ((v1.value << 4) + k0) ^ (v1.value + total.value) ^
((v1.value >> 5) + k1)
        total.value += delta

    return v0.value, v1.value


# test
if __name__ == "__main__":
    # 待加密的明文，两个32位整型，即64bit的明文数据
    value = [0, 0]
    buf2 = [0x2E63829D, 0xC14E40OF, 0x9B39BFB9, 1512016660, 0x61886DDE,
0x6565C6CF, 0x9FO64F64, 0x236A43F6]
    # buf2=[0x9D82632E,  0x0F404EC1,  0x9B39BFB9,  1512016660,  0x61886DDE,
0x6565C6CF,  0x9FO64F64,  0x236A43F6]
    # 四个key，每个是32bit，即密钥长度为128bit
    key = [0x12345678, 0x23456789, 0x34567890, 0x45678901]
    for i in range(0, len(buf2), 2):
        value[0], value[1] = buf2[i], buf2[i + 1]
        res = decrypt(value, key)
        bytearray.fromhex(hex(res[0])[2::]).decode()
        print(bytearray.fromhex(hex(res[0])[2::]).decode()[::-1],
bytearray.fromhex(hex(res[1])[2::]).decode()[::-1],
              sep='', end='')
    print('k}', end='')
# hgame{Tea_15_4_v3ry_h3a1thy_drlnk}
```

easyasm

```
a=
[0x5b,0x54,0x52,0x5e,0x56,0x48,0x44,0x56,0x5f,0x50,0x3,0x5e,0x56,0x6c,0x47,0x3,0
x6c,0x41,0x56,0x6c,0x44,0x5c,0x41,0x2,0x57,0x12,0x4e]
for i in a:
    print(chr(i^0x33),end='')# hgame{welc0me_t0_re_wor1d!}
```

easyenc

```
a = [0x04, 0xFF, 0xFD, 0x09, 0x01, 0xF3, 0xB0, 0x00, 0x00, 0x05, 0xF0, 0xAD,
0x07, 0x06, 0x17, 0x05, 0xEB, 0x17, 0xFD,
     0x17, 0xEA, 0x01, 0xEE, 0x01, 0xEA, 0xB1, 0x05, 0xFA, 0x08, 0x01, 0x17,
0xAC, 0xEC, 0x01, 0xEA, 0xFD, 0xF0, 0x05,
     0x07, 0x06, 0xF9]
for i in a:
    if i > 0xA0:
        i=-(0xff-i+1)
    print(chr((i + 86) ^ 0x32), end='')
# hgame{4ddit1on_is_a_rever5ible_0peration}
```

encode

```
v4=[0x00000008, 0x00000006, 0x00000007, 0x00000006, 0x00000001, 0x00000006,
0x0000000D, 0x00000006, 0x00000005, 0x00000006, 0x0000000B, 0x00000007,
0x00000005, 0x00000006, 0x0000000E, 0x00000006, 0x00000003, 0x00000006,
0x0000000F, 0x00000006, 0x00000004, 0x00000006, 0x00000005, 0x00000006,
0x0000000F, 0x00000005, 0x00000009, 0x00000006, 0x00000003, 0x00000007,
0x0000000F, 0x00000005, 0x00000005, 0x00000006, 0x00000001, 0x00000006,
0x00000003, 0x00000007, 0x00000009, 0x00000007, 0x0000000F, 0x00000005,
0x00000006, 0x00000006, 0x0000000F, 0x00000006, 0x00000002, 0x00000007,
0x0000000F, 0x00000005, 0x00000001, 0x00000006, 0x0000000F, 0x00000005,
0x00000002, 0x00000007, 0x00000005, 0x00000006, 0x00000006, 0x00000007,
0x00000005, 0x00000006, 0x00000002, 0x00000007, 0x00000003, 0x00000007,
0x00000005, 0x00000006, 0x0000000F, 0x00000005, 0x00000005, 0x00000006,
0x0000000E, 0x00000006, 0x00000007, 0x00000006, 0x00000009, 0x00000006,
0x0000000E, 0x00000006, 0x00000005, 0x00000006, 0x00000005, 0x00000006,
0x00000002, 0x00000007, 0x0000000D, 0x00000007, 0x00000000, 0x00000000,
0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
0x00000000, 0x00000000, 0x00000000, 0x00000000]
for i in range(50):
    print(chr((v4[2 * i + 1] << 4) | v4[2 * i]),end='')
# hgame{encode_is_easy_for_a_reverse_engineer}
```

test_your_IDA

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char Str1[24]; // [rsp+20h] [rbp-18h] BYREF

  sub_140001064("%10s");
  if ( !strcmp(Str1, "r3ver5e") )
    sub_140001010("your flag:hgame{te5t_y0ur_IDA}");
  return 0;
}
```

ida打开获得flag

hgame{te5t_y0ur_IDA}

BlockChain

Checkin

看看源码

```
// SPDX-License-Identifier: MIT

pragma solidity 0.8.17;

contract Checkin {
    string greeting;

    constructor(string memory _greeting)  {
        greeting = _greeting;
    }

    function greet() public view returns (string memory) {
        return greeting;
    }

    function setGreeting(string memory _greeting) public {
        greeting = _greeting;
    }

    function isSolved() public view returns (bool) {
        string memory expected = "HelloHGAME!";
        return keccak256(abi.encodePacked(expected)) ==
keccak256(abi.encodePacked(greeting));
    }
}
```

```
[+] deployer account: 0x5B9E13374B97E1B1633dfAa4c36A8OCA1655CA4a
[+] token:
v4.local.R3OAQ2ikgeB8XRLjt8bggKUnkbwMGMlv_mSRXwxmGUDuClnl2FqmeIZn1cZF9jGB8lW9Yny
I2GHlI2Tc_Z4s-auwOmQri4SpuBN3HmrPyoYJNoj5eHgCE93i5sw1jxQhkAk-
vc82sQCz82FTYgOElp93TYykIQR9EIItY49C66__Yg
[+] contract address: 0x34DOaCD466AOf2O8e57722D4aF8O479AO47B3271
[+] transaction hash:
0xd2a3e46e1dd2O1c49325a2c819568229dc318O1f655Oeb8db7a3af6c77796e93
```

简单的区块链题目，交互一下就有flag了

```
import json
import time
from web3 import Web3, HTTPProvider

contract_address = '0x34D0aCD466A0f208e57722D4aF80479A047B3271'
private_key = [你的钱包私钥]
wallet = Web3.toChecksumAddress([你的钱包地址])

w3 = Web3(HTTPProvider('http://week-1.hgame.lwsec.cn:32663'))

ABI = json.loads(
    '[{"inputs":
[{"internalType":"string","name":"_greeting","type":"string"}],"stateMutability"
:"nonpayable","type":"constructor"},{"inputs":[],"name":"greet","outputs":
[{"internalType":"string","name":"","type":"string"}],"stateMutability":"view","
type":"function"},{"inputs":[],"name":"isSolved","outputs":
[{"internalType":"bool","name":"","type":"bool"}],"stateMutability":"view","type
":"function"},{"inputs":
[{"internalType":"string","name":"_greeting","type":"string"}],"name":"setGreeti
ng","outputs":[],"stateMutability":"nonpayable","type":"function"}]')

# w3.eth.enable_unaudited_features()

contract = w3.eth.contract(address=contract_address, abi=ABI)
nonce = w3.eth.getTransactionCount(wallet)
gasPrice = w3.toWei('10', 'gwei')
gasLimit = 4000000
tx = {
    'nonce': nonce,
    'gas': gasLimit,
    'gasPrice': gasPrice,
    'from': wallet
}

transaction = contract.functions.setGreeting("HelloHGAME!").buildTransaction(tx)
signed_tx = w3.eth.account.sign_transaction(transaction, private_key)
tx_hash = w3.eth.sendRawTransaction(signed_tx.rawTransaction)
transaction_hash = w3.toHex(tx_hash)
tx_receipt = w3.eth.wait_for_transaction_receipt(transaction_hash)
```