Frankss 战队 WRITEUP

一、 战队信息

战队名称：**Frankss**

战队排名：11

二、 解题情况

三、 解题过程

1 问卷调查

问卷

2 babyre

操作内容：

混淆见过很多次但是不知道是什么名字的混淆，总之就是先做一堆变量名重命名替换同义词，技巧是在最后比较的地方打个 log，然后每次修改代码后检查固定输入的加密结果，相同就证明代码没改崩。

替换完同义词之后吧['XXX']这个文本在 F12 里模拟执行一下手动复原出来，然后改成.xxx

最后删去没引用的函数和变量，重命名一下剩下的变量语义就十分清晰了

简单解密一下 TEA，因为有两种可能，延迟函数执行前和执行后，所以两种都解密一下，区别就是
执行前 delta 值是 1，check[3]为 1700913016
执行后 delta 值是 10，check[3]为 0
然后发现后者得到的 flag 第 2 3 4 段写不在 0xffff 区间内，第一段的可以（转 int32 之后）
所以用执行前的值，做解密脚本

```
var delta = 1;
function encryptBlock(arr_a, arr_b, cnt, index) {
    var y = arr_a[index % arr_a.length];
    var z = arr_a[(index + 1) % arr_a.length];
    var sum = 0;
    for (var i = 0; i < cnt; i++) {
        sum += delta;
        y += ((z << 4) + arr_b[0] ^ z + sum ^ (z >>> 5) + arr_b[1]);
        z += ((y << 4) + arr_b[2] ^ y + sum ^ (y >>> 5) + arr_b[3]);
    }
    arr_a[index % arr_a.length] = y;
    arr_a[(index + 1) % arr_a.length] = z;
}
function decryptBlock(arr_a, arr_b, cnt, index) {
    var y = arr_a[index % arr_a.length];
    var z = arr_a[(index + 1) % arr_a.length];
    var sum = cnt*delta;
    for (var i = 0; i < cnt; i++) {
        z -= ((y << 4) + arr_b[2] ^ y + sum ^ (y >>> 5) + arr_b[3]);
        y -= ((z << 4) + arr_b[0] ^ z + sum ^ (z >>> 5) + arr_b[1]);
        sum -= delta;
    }
```

```
        arr_a[index % arr_a.length] = y;
        arr_a[(index + 1) % arr_a.length] = z;
    }
    function check() {
        var flag_in =  [3256077785, 2046034498, 844956517, 1700913016,
1464334280, 3644773429];
        for (var i = flag_in.length-1;i>=0; i -= 1) {
            decryptBlock(flag_in, [
                flag_in[(2 + i) % flag_in.length],
                flag_in[(3 + i) % flag_in.length],
                flag_in[(4 + i) % flag_in.length],
                flag_in[(5 + i) % flag_in.length]
            ], 34, i);
        }
        var flag_enc = new Uint32Array(flag_in.length);
        for (var i = 0; i < flag_in.length; i += 1) {
            flag_enc[i] = flag_in[i];
        }
        console.log(flag_enc);

    }
```

执行后的结果转 hex 为 847213d0-57a8-4411-b0f8-c036474-deadbeef，根据 flag 格式补一下为 847213d0-57a8-4411-b0f8-000000c036474

flag 值：

flag{847213d0-57a8-4411-b0f8-000000c036474}