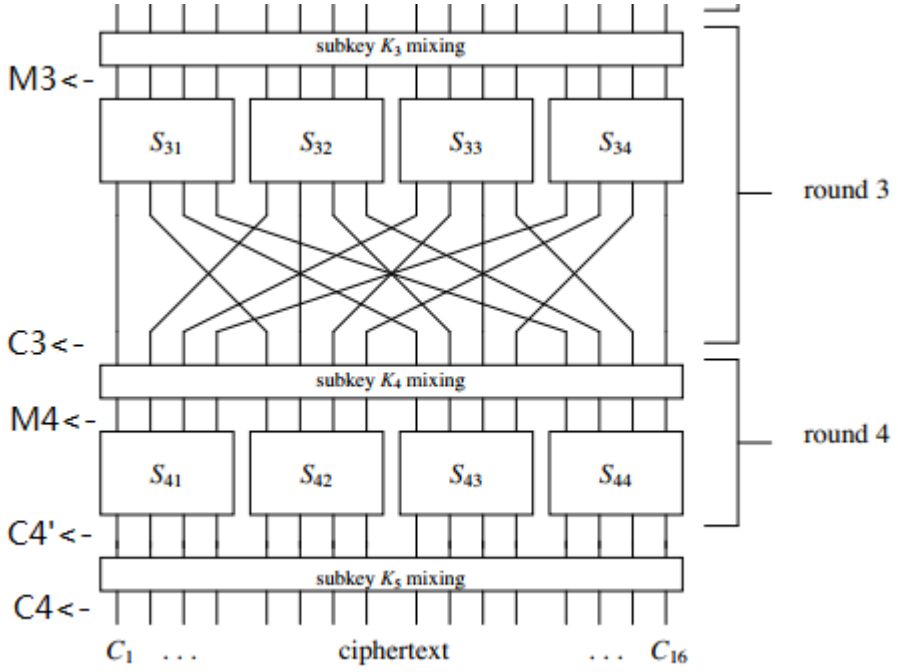
1901210530 吴宏凯

以 Basic SPN 为例，回答为什么加密算法最后一轮后还需要做一次密钥混合？

结论：如果没有最后一轮的子密钥混合，分析者可以直接对最后一轮的 S-box 做逆运算，从而忽略最后一轮的加密作用。

证明：

下图为 SPN 的第三、第四轮加密：



设每一轮完整加密都依次包括密钥混合，S 置换，P 调换（第 4 轮没有），密钥混合（第 4 轮才有）；

设每一轮的输入分别为 M1,M2,M3,M4，每一轮完整加密的输出为 C1,C2,C3,C4；则 C3 与 K4 做密钥混合可得到第四轮的输入 M4，即 C3^K4=M4；

去掉最后一轮密钥混合后得到的输出密文为 C4'，即 C4'^K5=C4；

若此时去掉最后一轮密钥混合，加密算法最终得到的是 C4'，则此时借助逆 S4 的表格能直接求出 M4，相当于第四轮加密无效。

实例：借助 NaiveSPN 加解密代码

输入的明文 M1=492e,

计算前三轮加密后的输出：C3=9641，再与 K4 做一次密钥混合得到 M4=7953

去掉最后一轮密钥混合后得到 C4'=8af1

此时计算 C4'在 SInv[16]中的置换，得到 X=7953=M4。

得证。

附 C4'在 SInv 中做置换的函数：

```
unsigned short naivespn_withoutk5_decrypt(unsigned short c){
    unsigned short s = c;
    unsigned int x0, x1, x2, x3;
    /* substitution */
```

```
    x0 = SInv[s & 0xf];
    x1 = SInv[(s >> 4) & 0xf];
    x2 = SInv[(s >> 8) & 0xf];
    x3 = SInv[(s >> 12) & 0xf];
    s = x0 | (x1 << 4) | (x2 << 8) | (x3 << 12);
    return s;
}
```