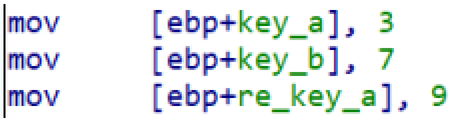
仿射密码逆向

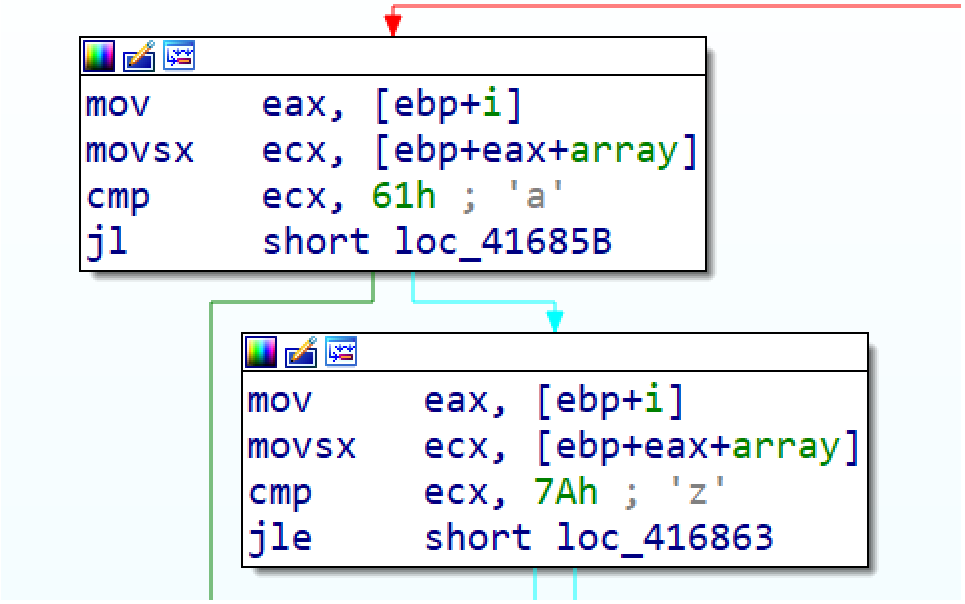
逻辑分析

1. 定义密钥



```
mov     [ebp+key_a], 3
mov     [ebp+key_b], 7
mov     [ebp+re_key_a], 9
```

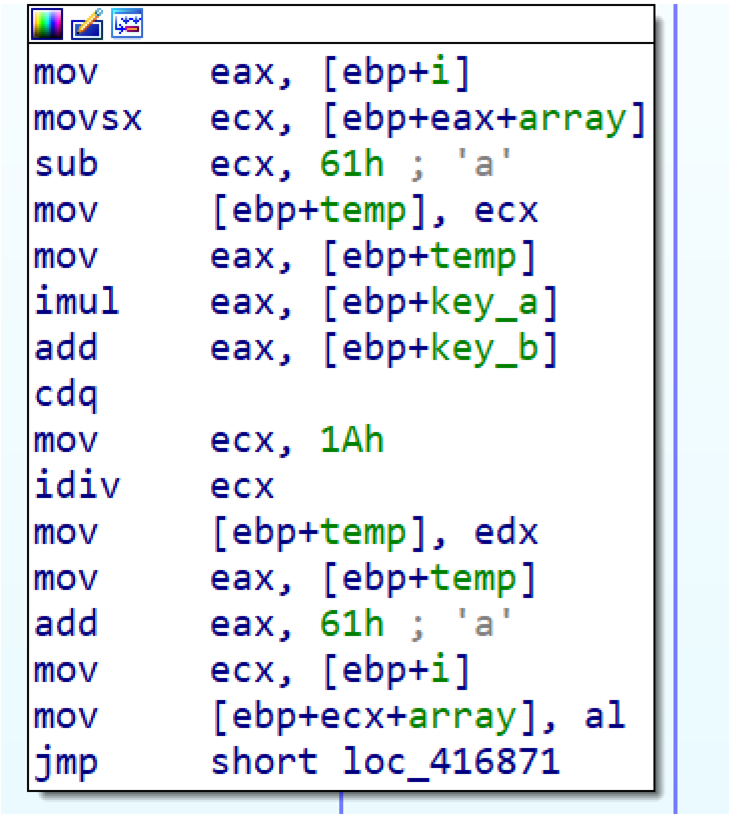
2. 范围判定： a 和 z 之间进入加密流程；



```
mov     eax, [ebp+i]
movsx   ecx, [ebp+eax+array]
cmp     ecx, 61h ; 'a'
jl      short loc_41685B
```

```
mov     eax, [ebp+i]
movsx   ecx, [ebp+eax+array]
cmp     ecx, 7Ah ; 'z'
jle     short loc_416863
```

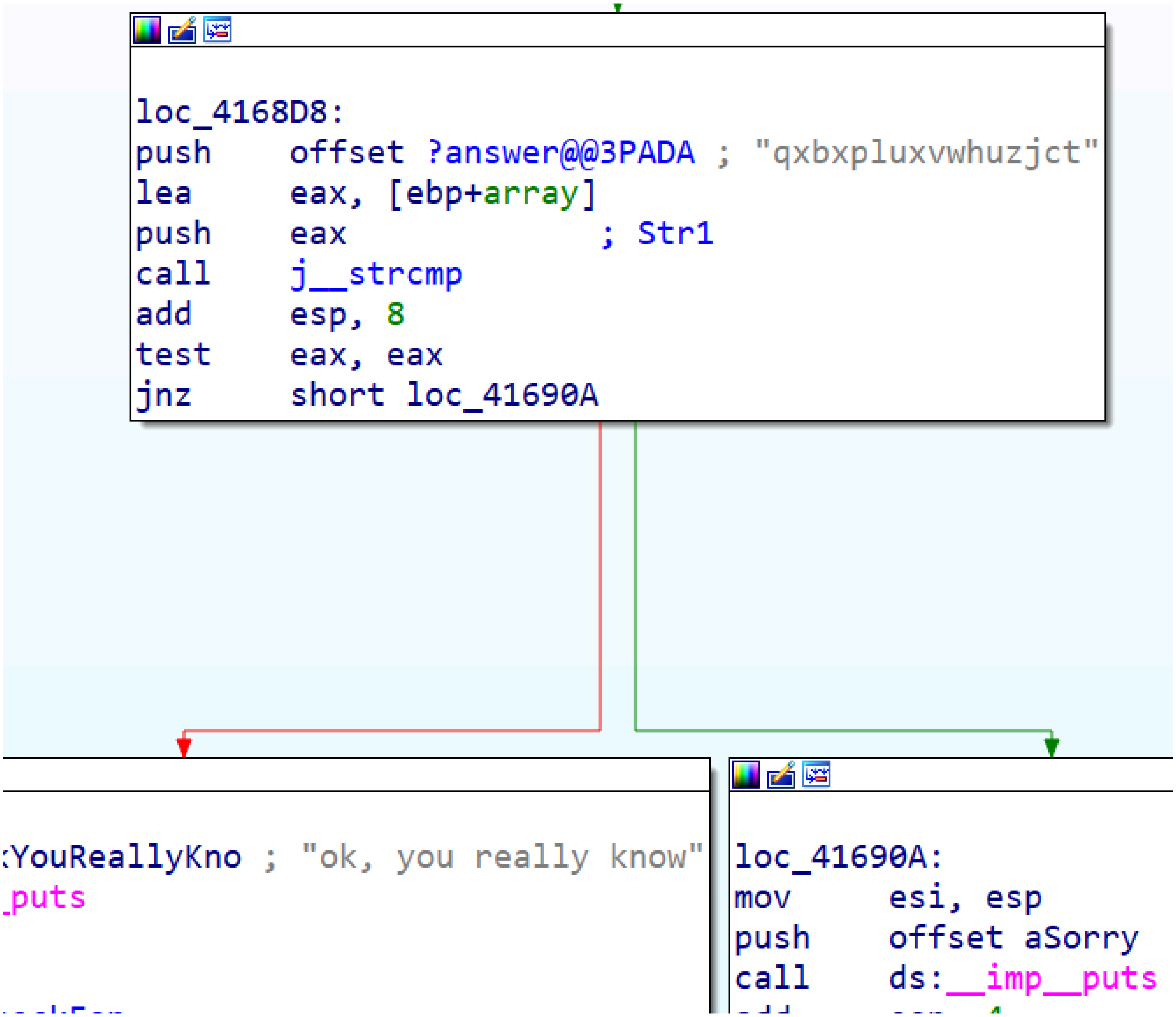
3. 加密：



```
mov     eax, [ebp+i]
movsx   ecx, [ebp+eax+array]
sub     ecx, 61h ; 'a'
mov     [ebp+temp], ecx
mov     eax, [ebp+temp]
imul    eax, [ebp+key_a]
add     eax, [ebp+key_b]
cdq
mov     ecx, 1Ah
idiv    ecx
mov     [ebp+temp], edx
mov     eax, [ebp+temp]
add     eax, 61h ; 'a'
mov     ecx, [ebp+i]
mov     [ebp+ecx+array], al
jmp     short loc_416871
```

- 逐字符加密，先将字符减 a 即 61h(97)；
- 然后计算 key_a * x + b 存到 temp 中;
- temp 除以 26 的余数（edx 中）存入 temp；
- temp + 61h 结果传入数组对应位置，完成当前字符加密。

4. 结果判定：加密结果和 answer 比较，相同则通过。



```
loc_4168D8:
push    offset ?answer@@3PADA ; "qxbxpluxvwhuzjct"
lea     eax, [ebp+array]
push    eax             ; Str1
call    j__strcmp
add     esp, 8
test    eax, eax
jnz     short loc_41690A
```

```
:YouReallyKno ; "ok, you really know"
_puts
```

```
loc_41690A:
mov     esi, esp
push    offset aSorry
call    ds:__imp__puts
```