

```

lea     edx, [ebx+13h]
add     edi, 3
mov     [esp+4Ch+var_20], edx
mov     [esp+4Ch+dest], edx
mov     [esp+4Ch+n], 0A14h
mov     [esp+4Ch+src], offset aT1_lib_c; "t1_lib.c"
call    CRYPTO_malloc
mov     byte ptr [eax], 2
mov     ebp, eax
mov     eax, ebx
shr     eax, 8
mov     [ebp+2], b1
lea     edx, [ebp+3]
mov     [ebp+1], a1
mov     [esp+4Ch+n], ebx; n
mov     [esp+4Ch+dest], edx; dest
mov     [esp+4Ch+var_24], edx
mov     [esp+4Ch+src], edi; src
call    _memcpy ← call memcpy
mov     edx, [esp+4Ch+var_24]
mov     [esp+4Ch+src], 10h
add     ebx, edx
mov     [esp+4Ch+dest], ebx
call    RAND_pseudo_bytes
mov     edx, [esp+4Ch+var_20]
mov     [esp+4Ch+n], ebp
mov     [esp+4Ch+src], 18h
mov     [esp+4Ch+dest], esi
mov     [esp+4Ch+var_40], edx
call    ssl3_write_bytes
test    eax, eax
js      short loc_80B7920

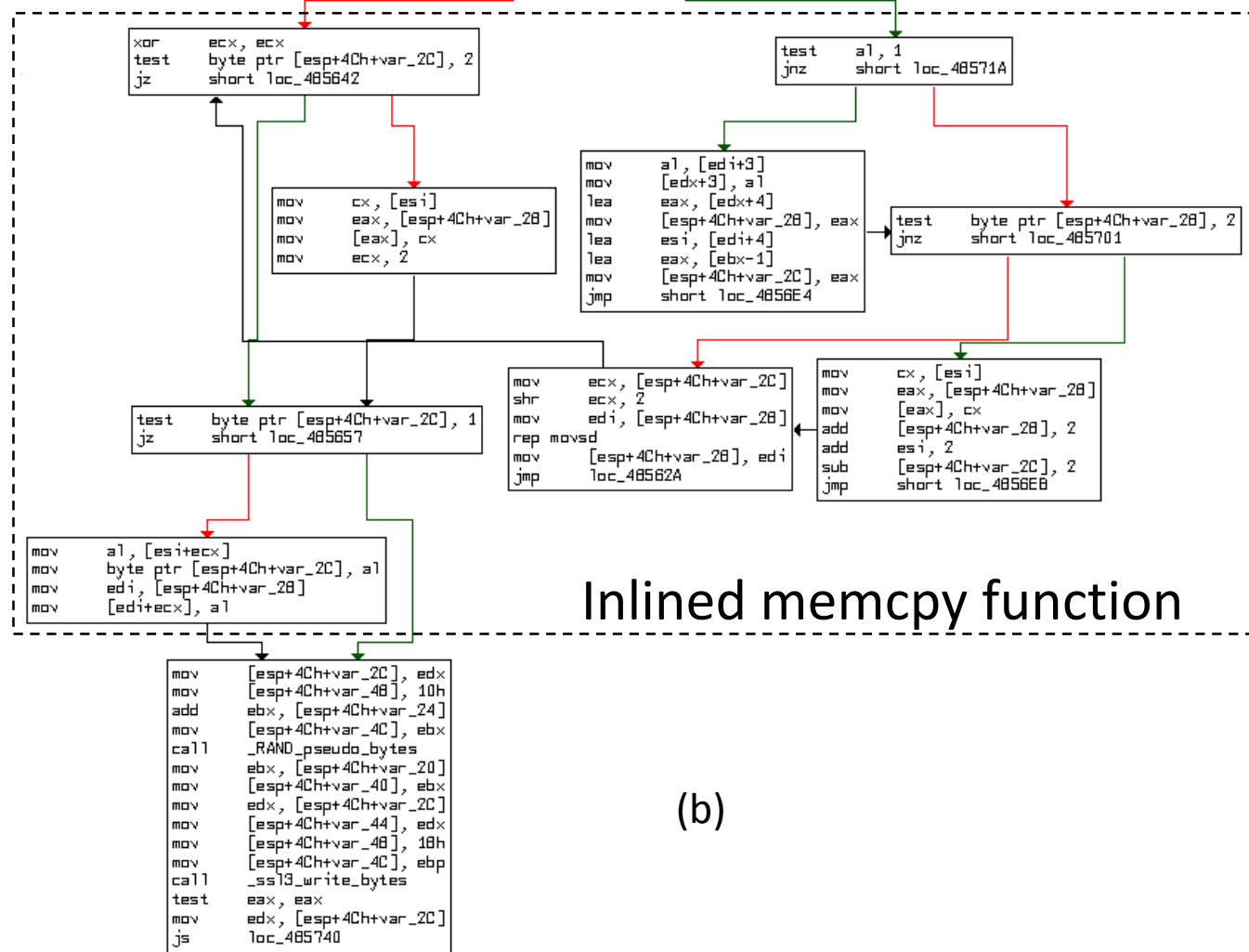
```

(a)

```

lea     eax, [ebx+13h]
mov     [esp+4Ch+var_20], eax
mov     [esp+4Ch+var_44], 0A14h
mov     [esp+4Ch+var_48], offset aT1_lib_c; "t1_lib.c"
mov     [esp+4Ch+var_4C], eax
call    _CRYPTO_malloc
mov     edx, eax
mov     byte ptr [eax], 2
mov     eax, ebx
shr     eax, 8
mov     [edx+1], a1
mov     [edx+2], b1
lea     eax, [edx+3]
mov     [esp+4Ch+var_24], eax
mov     [esp+4Ch+var_2C], ebx
mov     [esp+4Ch+var_28], eax
lea     esi, [edi+3]
cmp     ebx, 4
jnb     loc_4B56E0

```



Inlined memcpy function

(b)