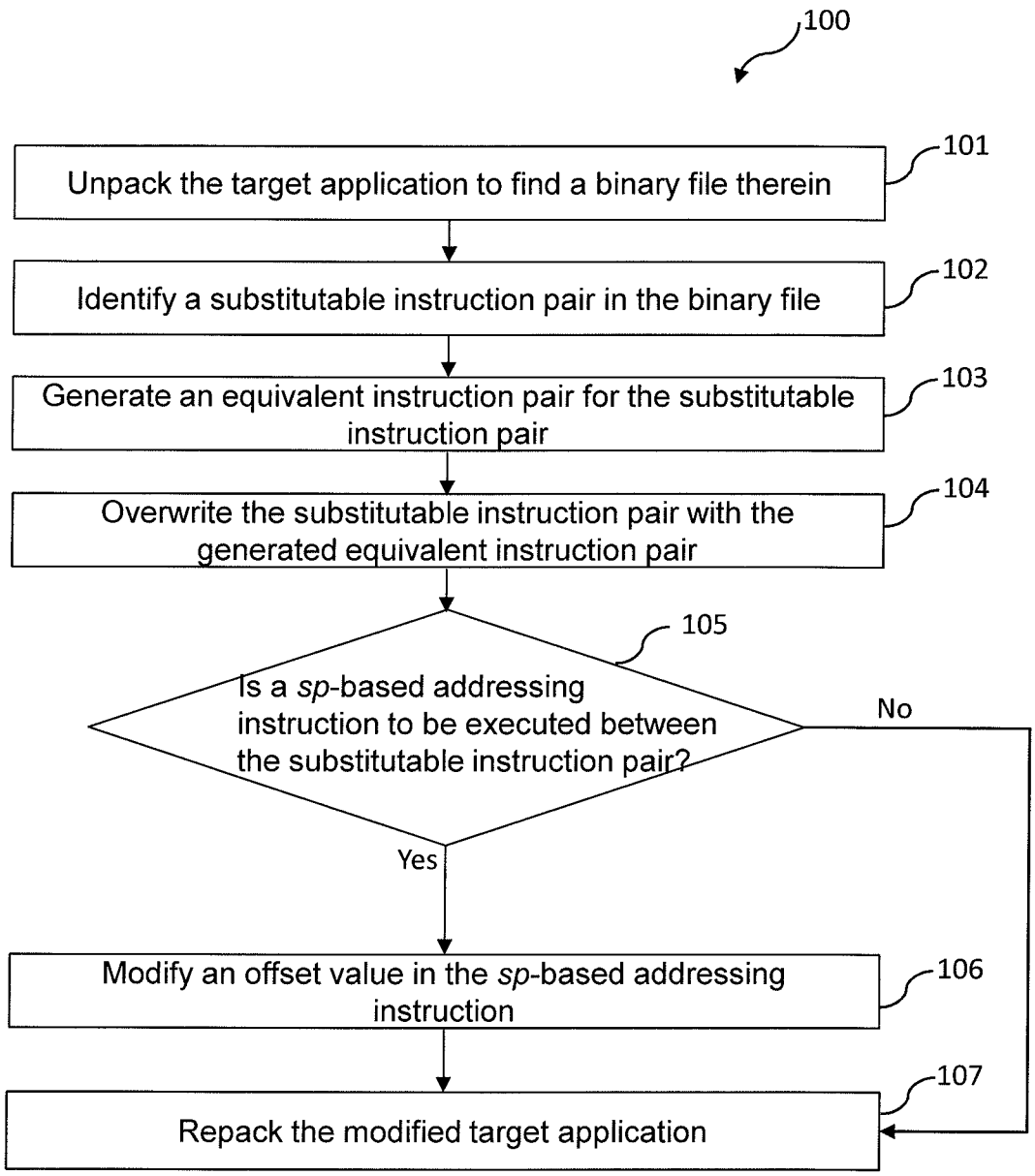
100
Unpack the target application to find a binary file therein
101
Identify a substitutable instruction pair in the binary file
102
Generate an equivalent instruction pair for the substitutable instruction pair
103
Overwrite the substitutable instruction pair with the generated equivalent instruction pair
104
Is a sp-based addressing instruction to be executed between the substitutable instruction pair?
105
No
Yes
Modify an offset value in the sp-based addressing instruction
106
Repack the modified target application
107

Figure 1

```
                set_linebreaks_utf16:
000015cc            push        {r4, r5, r6, r7, lr}
000015ce            sub.w       sp, sp, #0x800
000015d2            sub         sp, #0xc
000015d4            mov         r5, r0
000015d6            mov         r4, r1
000015d8            mov         r6, r2
000015da            add         r0, sp, #0x8
000015dc            lsls        r2, r1, #0x1
000015de            mov         r1, r5
000015e0            mov         r7, r3
000015e2            blx         memcpy@PLT
000015e6            ldr         r3, = 0x39b8
000015e8            mov         r0, r5
000015ea            mov         r1, r4
000015ec            add         r3, pc
000015ee            ldr         r3, [r3]
000015f0            mov         r2, r6
000015f2            str         r3, [sp]
000015f4            mov         r3, r7
000015f6            bl          set_linebreaks
000015fa            add         sp, #0xc
000015fc            add.w       sp, sp, #0x800
00001600            pop         {r4, r5, r6, r7, pc}
                        ; endp
--------
```

Figure 2A

```
                set_linebreaks_utf16:
000015cc            push        {r1, r2, r3, r4, r5, r6, r7, lr}
000015ce            sub.w       sp, sp, #0x800
000015d2            sub         sp, #0xc
000015d4            mov         r5, r0
000015d6            mov         r4, r1
000015d8            mov         r6, r2
000015da            add         r0, sp, #0x8
000015dc            lsls        r2, r1, #0x1
000015de            mov         r1, r5
000015e0            mov         r7, r3
000015e2            blx         memcpy@PLT
000015e6            ldr         r3, = 0x39b8
000015e8            mov         r0, r5
000015ea            mov         r1, r4
000015ec            add         r3, pc
000015ee            ldr         r3, [r3]
000015f0            mov         r2, r6
000015f2            str         r3, [sp]
000015f4            mov         r3, r7
000015f6            bl          set_linebreaks
000015fa            add         sp, #0xc
000015fc            add.w       sp, sp, #0x800
00001600            pop         {r1, r2, r3, r4, r5, r6, r7, pc}
                        ; endp
```

Figure 2B

```
                sub_49298:
00049298            push        {r4, lr}
0004929c            ldr         r1, [sp, #0xc]
000492a0            mov         r4, r0
000492a4            bl          sub_45130
000492a8            mov         r0, r4
000492ac            pop         {r4, pc}
```

Figure 3A

```
                sub_49298:
00049298            push        {r4, r6, lr}
0004929c            ldr         r1, [sp, #0x10]
000492a0            mov         r4, r0
000492a4            bl          sub_45130
000492a8            mov         r0, r4
000492ac            pop         {r4, r6, pc}
                    ; endp
```

Figure 3B

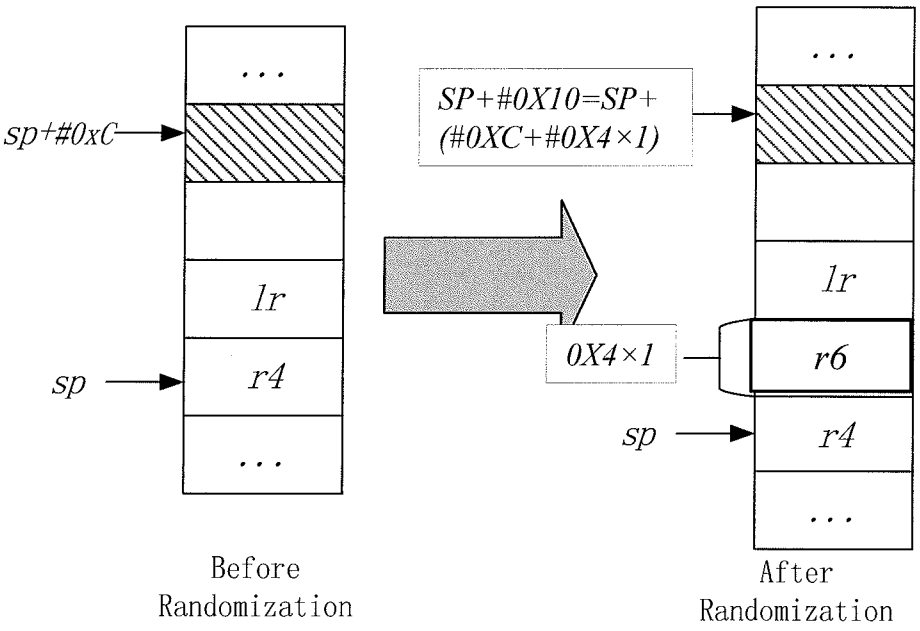

Figure 3C

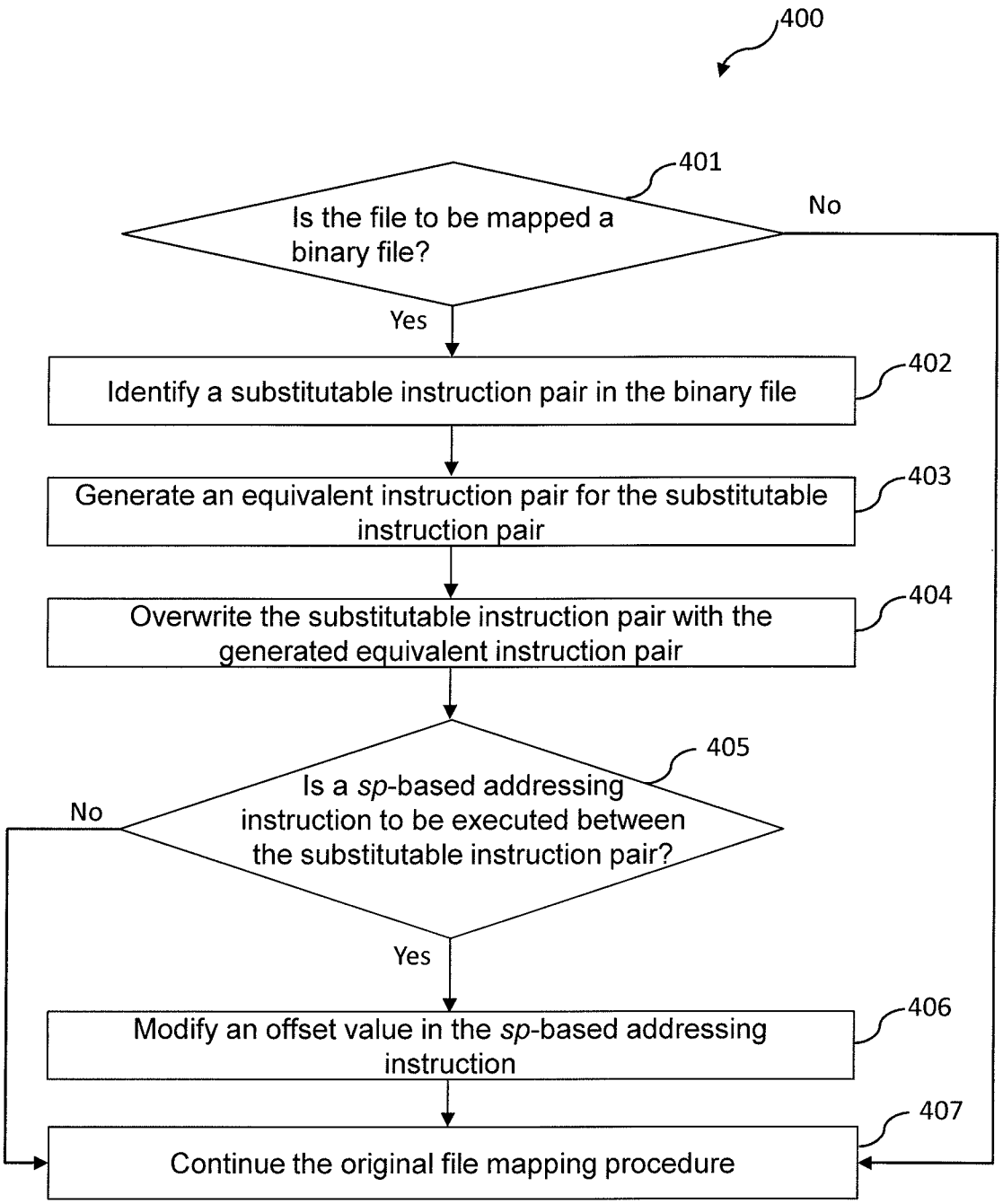
400
401
Is the file to be mapped a binary file?
No
Yes
402
Identify a substitutable instruction pair in the binary file
403
Generate an equivalent instruction pair for the substitutable instruction pair
404
Overwrite the substitutable instruction pair with the generated equivalent instruction pair
405
Is a sp-based addressing instruction to be executed between the substitutable instruction pair?
No
Yes
406
Modify an offset value in the sp-based addressing instruction
407
Continue the original file mapping procedure

Figure 4