

First, we investigate, `this.b.a().h().a((byte)0x80)`. We see that `this.d` is a byte and the `toString()` implies it's the CID.

```
public class b {
    // Some code omitted for brevity
    private byte d;

    public void a(byte b) {
        this.d = b;
    }

    @Override
    public String toString() {
        return "CryptogramOutput{ATC=" + this.a.c() + ", issuerApplicationData=" + this.b.c() +
               ", cryptogram=" + this.c.c() + ", cid=" + ((int)this.d) + '}';
    }
}
```

For `this.b.a().j().a((byte)0x80)`, we don't see anything meaningful or CID, so we'll ignore this for now.

```
public void a(byte b) {  
    this.h = b;  
}
```

Let's edit `this.b.a().h().a((byte)0x80)` to `this.b.a().h().a((byte)0x40)` in the Smali code, which is very simple after identifying the location.

const/16 v2, 0x40 # original value -0x80

invoke-virtual {v1, v2}, Lcom/b;->a(B)V

```
75     const/16 v2, 0x40
76
77     invoke-virtual {v1, v2}, Lcom/.../B; >> a(B)V
```

Let's double check in JEB (or any other decompiler like JADX) to see that **0x80** was changed to **0x40**. Success!

<pre> 00000026  igure-object    v1, p0, b-&gt;b:h 0000002A  invoke-virtual   h-&gt;a()a, v1 00000030  move-result-object v1 00000032  invoke-virtual   a-&gt;h()b, v1 00000038  move-result-object v1 0000003A  const/16         v2, 0x0040 0000003E  invoke-virtual   b-&gt;a(B)V, v1, v2 00000044  igure-object    v1, p0, b-&gt;b:h 00000048  invoke-virtual   h-&gt;a()a, v1 0000004E  move-result-object v1 00000050  invoke-virtual   a-&gt;j()a, v1 00000056  move-result-object v1 00000058  invoke-virtual   a-&gt;a(B)V, v1, v2 </pre>	<pre> private c a(com.██████████.a.c a(com.██████████) utils.a.a.d d0) {     int v = d0.a();     int v1 = this.e.a(0);     this.e.a(0, ((byte)(v1   0xFFFFFA0)));     this.b.a().h().a(0x40);     this.b.a().j().a(0x40);     if((v &amp; 16) == 16) {         int v2 = this.e.a(1);         this.e.a(1, ((byte)(v2   0x40)));     }     return this.b(d0); } </pre>
--	--

Note that **(byte)64** and **(byte)0x40** are the same and both equal to 64 in decimal.

```

private com.██████████.a.c a(com.██████████) utils.a.a.d dVar) {
    byte a2 = dVar.a();
    this.e.a(0, (byte) (this.e.a(0) | (-96)));
    this.f8193b.a().h().a((byte) 64);
    this.f8193b.a().j().a((byte) 64);
    if ((a2 & 16) == 16) {
        this.e.a(1, (byte) (this.e.a(1) | 64));
    }
    return b(dVar);
}

```