Unsym

unsym

这题 `go_parser` 可以恢复符号，虽然其最多支持到go1.18，但是go1.20有一部分结构和go1.18的 `firstmoduledata` 结构是相似的

步骤如下：

首先go编译的exe没有办法直接通过 `gopclntab` 段来定位特征码，可以通过搜索字符串 `go:buildid` 搭配交叉引用来定位，需要将0xFFFFFFF1改为0xFFFFFFF0(或者直接改脚本中的特征码)

```
.rdata:00000000004EB6F0 00 00 00 00 00 00 00 00 00 00+align 20h
.rdata:00000000004EB700 F0 FF FF FF                   runtime_symtab dd 0FFFFFFF0h
.rdata:00000000004EB700
.rdata:00000000004EB704 00 00                         dw 0
.rdata:00000000004EB706 01                            db    1
```

之后直接运行脚本即可恢复符号

这一段是对key进行RSA加密(65537以及exp函数可以大致猜出，刚开始被卡住了😭)，通过 `yafu` 可以分解 `n`

```
118  }
119  v45 = 65537LL;
120  v76[0] = 0;
121  v77 = &v45;
122  v78 = 1LL;
123  v79 = 1LL;
124  v73 = 0;
125  v74 = 0LL;
126  v75 = v4;
127  v70 = 0;
128  v71 = 0LL;
129  v72 = v4;
130  math_big__Int_SetString(16LL, v8, (__int64)&v45, v8);
131  math_big__Int_exp(v67, 0LL, v14, (__int64)v76);
132  v15 = v73;
133  v16 = 16LL;
134  v17 = math_big_nat_itoa(v73, 16LL);
135  v51 = runtime_slicebytetostring();
136  runtime_makeslice();
```

写脚本解出key： `E@sy_RSA_enc7ypt`

```
1 import libnum
2 from Crypto.Util.number import long_to_bytes, bytes_to_long
```

```
3
4  n = 0x1D884D54D21694CCD120F145C8344B729B301E782C69A8F3073325B9C5
5  c = 0xFAD53CE897D2C26F8CAD910417FBDD1F0F9A18F6C1748FACA10299DC8
6  q = 211549048872157489492804106164784 23
7  p = 376363184577451672341408081301567 39
8  e = 65537
9  d = libnum.invmod(e, (p - 1) * (q - 1))
10 m = pow(c, d, n)
11 string = long_to_bytes(m)
12 print(string)
```

之后将key作为AES_KEY和AES_IV对message文件进行AES-CBC加密

```
193   if ( !v30 )
194     runtime_panicdivide();
195   v39 = v30 - 7 % v30;
196   runtime_makeslicecopy(v37);
197   for ( j = 7LL; j < v39 + 7; ++j )
198     *(_BYTE *)(v31 + j) = v39;
199   v57 = v31;
200   v33 = (*(__int64 (**)(void))(v58 + 24))();
201   if ( v33 > v41 )
202 LABEL_23:
203     runtime_panicSliceAcap();
204   v34 = v55;
205   v40 = crypto_cipher_NewCBCEncrypter(v33);
206   v52 = v34;
207   v56 = runtime_makeslice();
208   ((void (*)(void))v40[4])();
209   if ( os_WriteFile() )
210   {
211     runtime_gopanic(v38);
212     goto LABEL_23;
213   }
```

解密文件，解密得到exe文件，运行即可得到flag

Recipe

AES Decrypt

Key
E@sy_RSA_enc7ypt
LATIN1

IV
E@sy_RSA_enc7ypt
LATIN1

Mode
CBC

Input
Raw

Output
Hex

Input

length: 104,048

Name: encrypted.bin

Size: 104,048 bytes

Type: application/octet-stream

Loaded: 100%

Output

start: 0 time: 21ms
end: 5849 length: 208090
length: 5849 lines: 1

4d5a90000300000004000000ffff0000b8000000000000004000000000000000000000000000000000000000
0000000000000000000000000800000000e1fba0e00b409cd21b8014ccd21546869732070726f6772616d2
2062652072756e20696e20444f53206d6f64652e0d0d0a2400000000000000504500004c010f006ac955
40000e00007010b0102180018000000120000000600000e0140000001000000030000000040000010000
0000010000000400000000000000d0010000040000026b401000300000000020000010000000001000
0001000000000000000000000000600000680500000000000000000000000000000000000000000000