

### What you need:

0. The BIOS firmware file of your computer. It can be downloaded from the official website or backed up from the computer. The detail can be searched online.

1. UEFITool, download address: <https://github.com/LongSoft/UEFITool/releases>

Note: Do not download the NE version, otherwise the BIOS content cannot be modified.

2. IDA, download address:

[https://www.hex-rays.com/products/ida/support/download\\_freeware/](https://www.hex-rays.com/products/ida/support/download_freeware/)

Note: The free version will be enough

3. Programmer (optional). If the modification fails and the system cannot boot, you can use the programmer to repair it. When purchasing a programmer, please check your own BIOS chip's voltage and make sure that the purchased programmer supports this chip. Many Windows tablets are using 1.8V low-voltage BIOS chip, the general programmer cannot be used directly, you need to pay attention when buying.

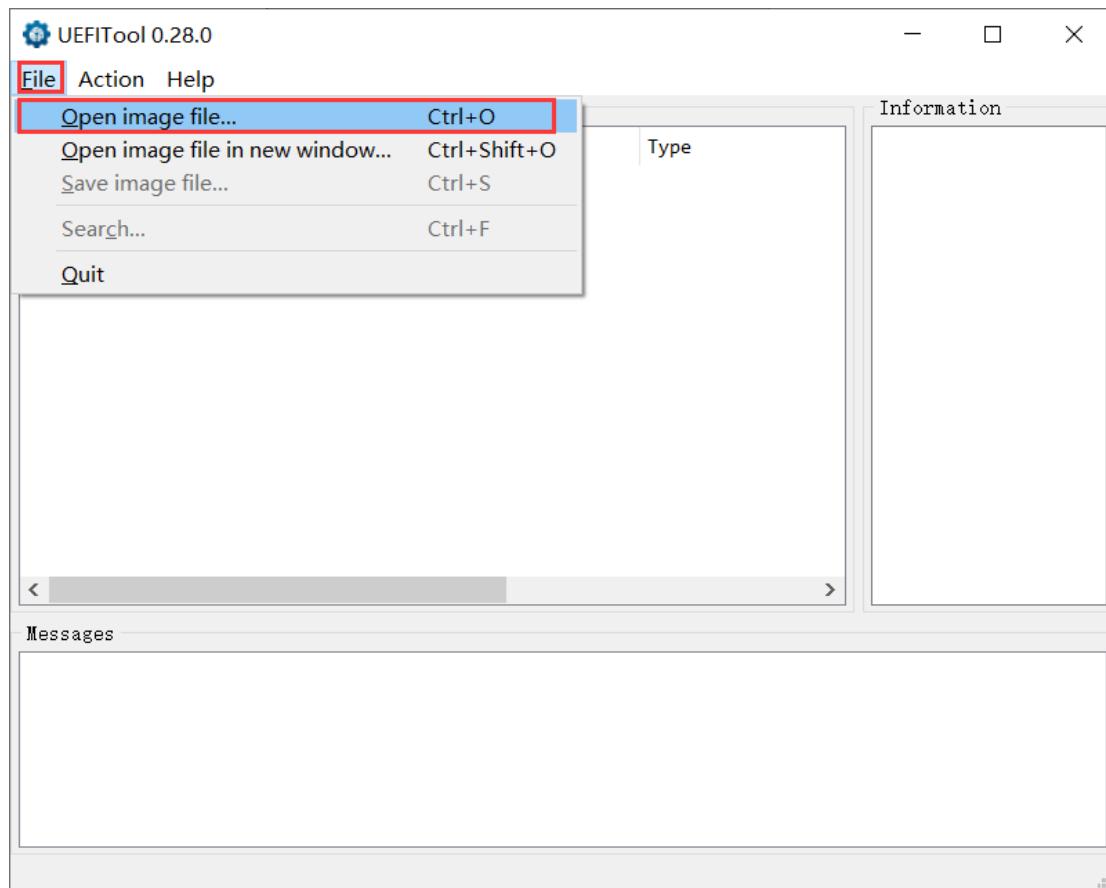
### Step one. Check the BIOS file

Before cracking, first make sure that your BIOS file is not encrypted. One of the easiest ways is to check the file size. If it is larger than 8MB, it is usually the encrypted BIOS.

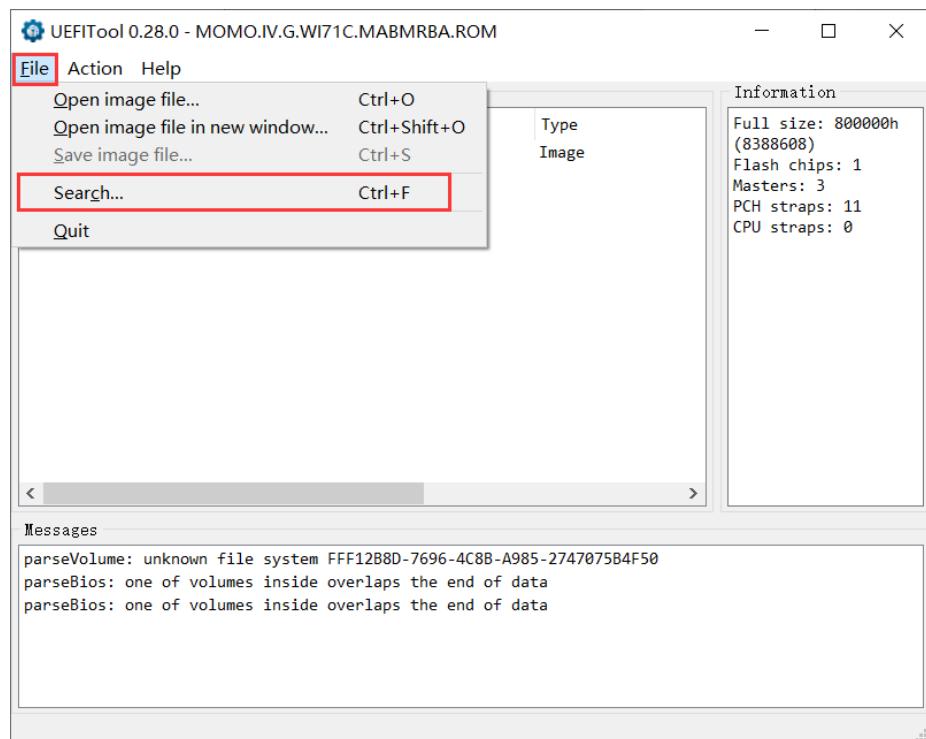


## Step two, extract the PEI module where the verification program is located

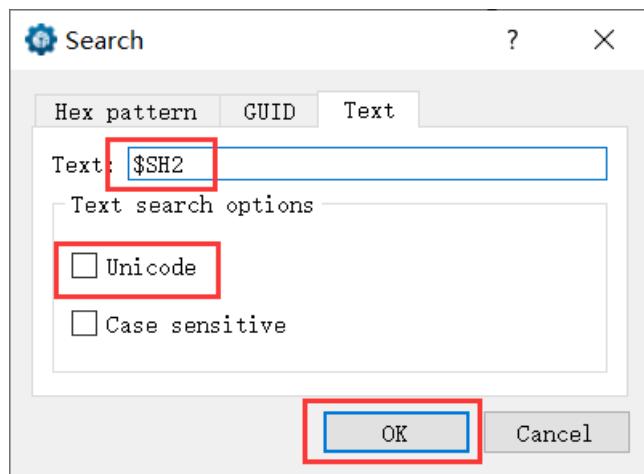
1. Use UEFITool to open (File-> Open image file) the BIOS to be modified



2. After loading the BIOS, select File-> Search to find the PEI module where the verification program is located.



3. Enter \$SH2, uncheck Unicode, and click OK.

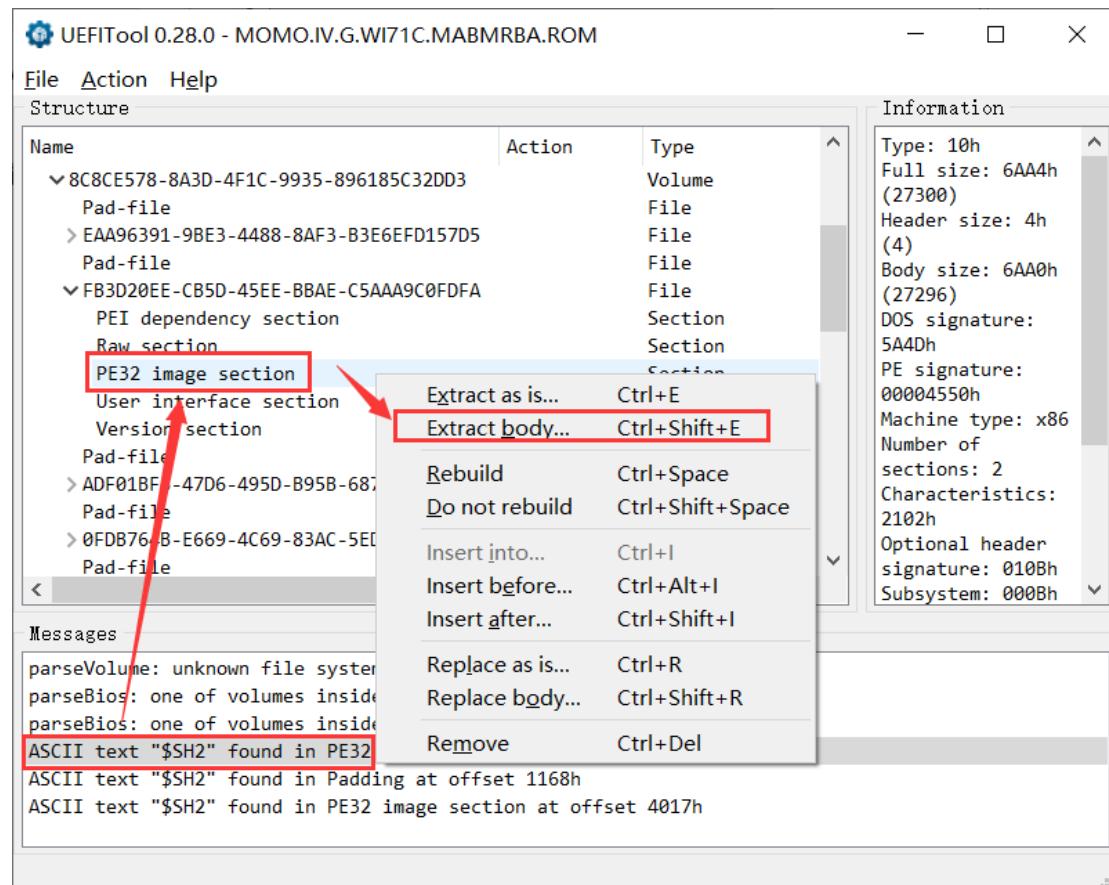


4. You should be able to see the three search results as shown in the figure below, of which there are two (maybe only one) PE32 images, and another one is the padding area. If not, please modify it carefully. Another possibility is that your BIOS uses other verification identifiers (such as \$HSS or \$SIG, etc.), you can use your BIOS's unique identifier to find the verification module.

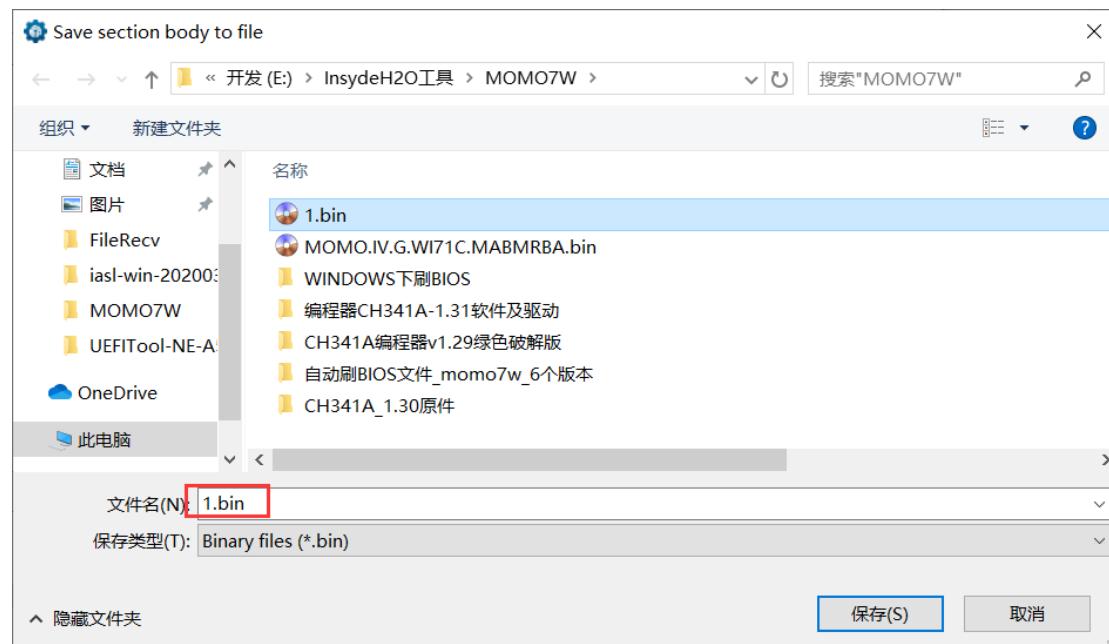
The screenshot shows the UEFITool 0.28.0 interface with the following details:

- Title bar: UEFITool 0.28.0 - MOMO.IV.G.WI71C.MABMRBA.ROM
- Menu bar: File, Action, Help
- Structure pane: Shows a single entry: Intel image.
- Information pane: Displays BIOS configuration details:
  - Full size: 800000h (8388608)
  - Flash chips: 1
  - Masters: 3
  - PCH straps: 11
  - CPU straps: 0
- Messages pane:
  - parseVolume: unknown file system FFF12B8D-7696-4C8B-A985-2747075B4F50
  - parseBios: one of volumes inside overlaps the end of data
  - parseBios: one of volumes inside overlaps the end of data
  - ASCII text "\$SH2" found in PE32 image section at offset 5E47h
  - ASCII text "\$SH2" found in Padding at offset 1168h
  - ASCII text "\$SH2" found in PE32 image section at offset 4017h

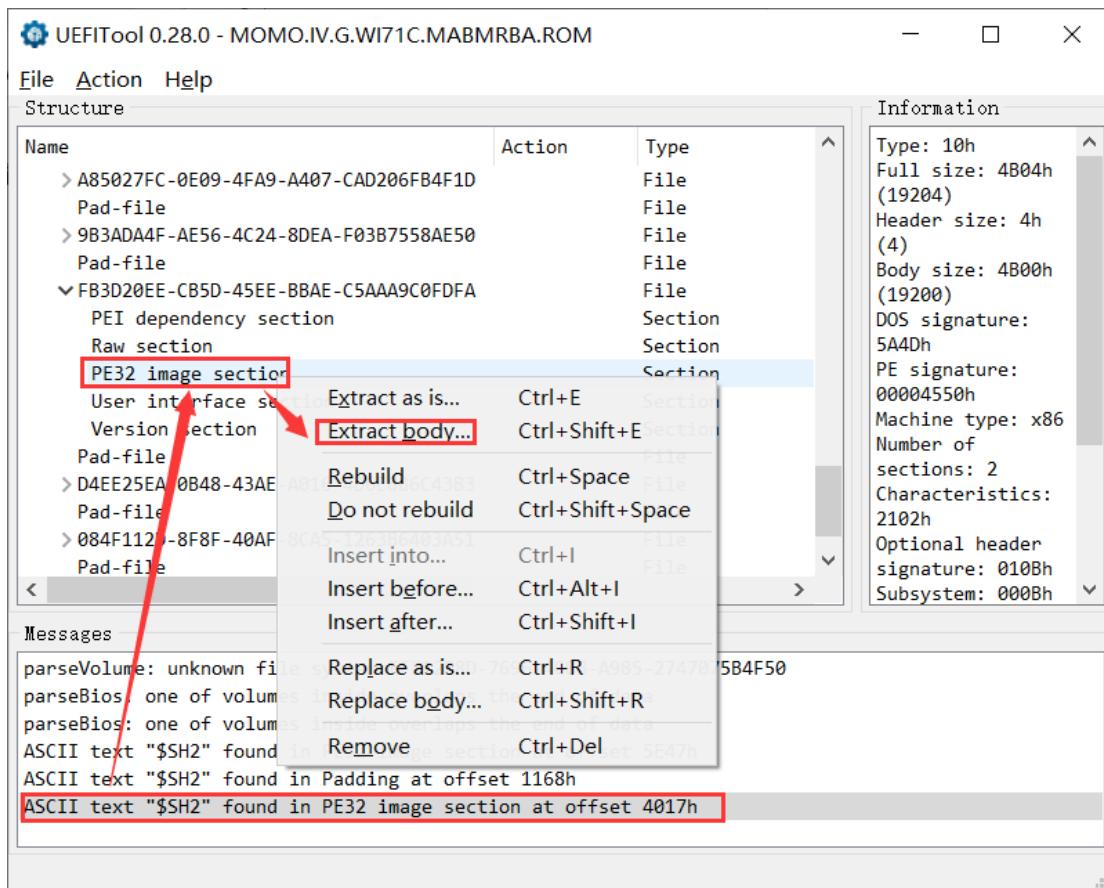
5. Double-click the first found PE32 image, right-click the PE32 image section above and select Extract body.



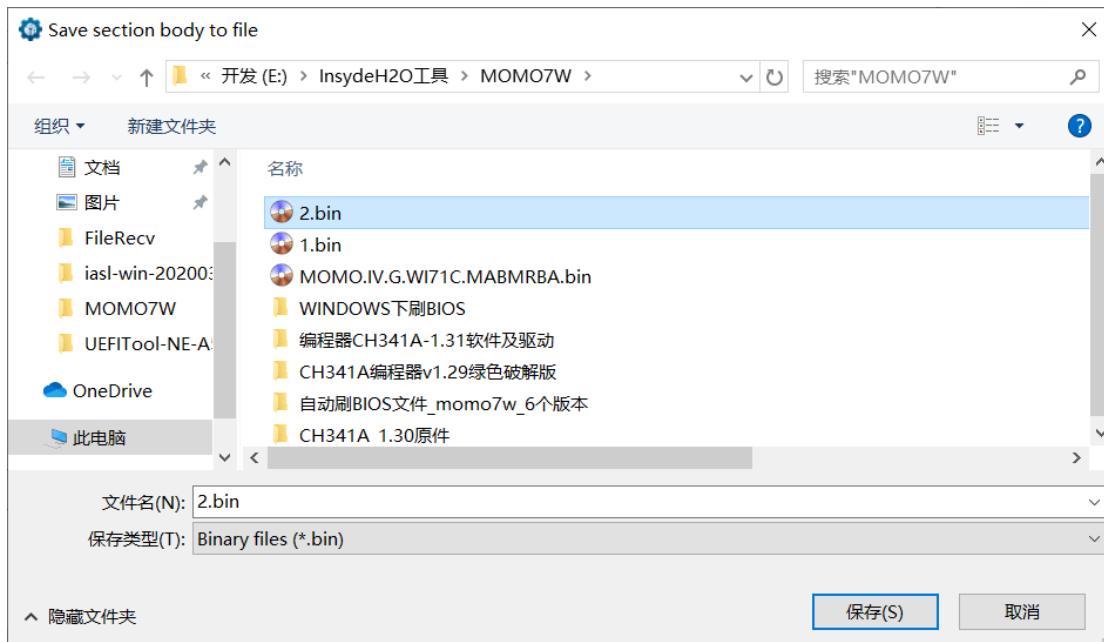
Save as 1.bin



6. Repeat the above operation, double-click the second searched PE32 image, and right-click to export the body.

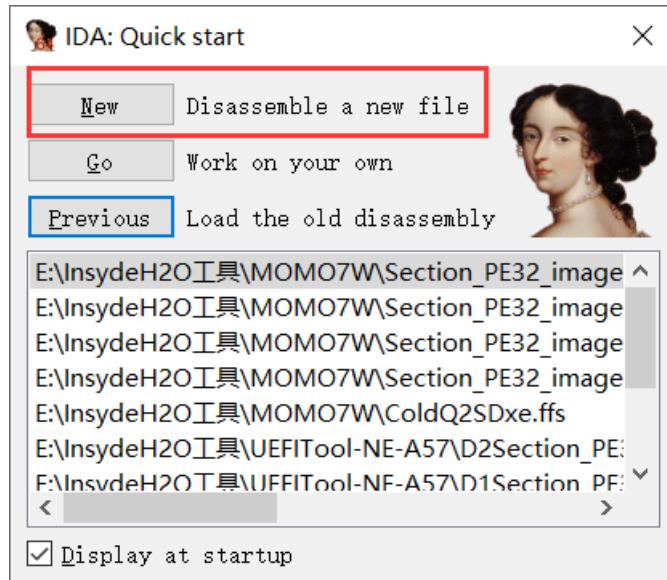


Save as 2.bin

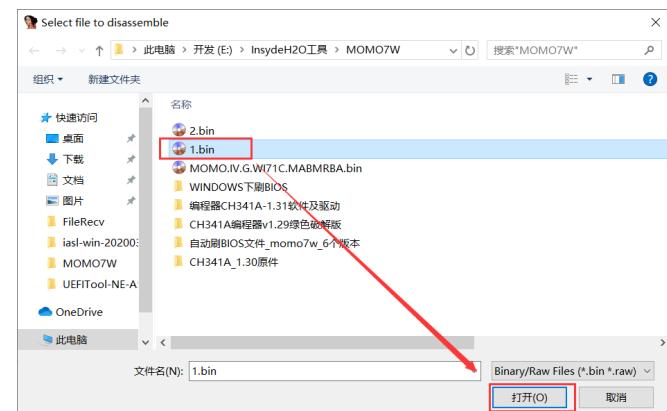


### Step three. Decompile the verification PEI module

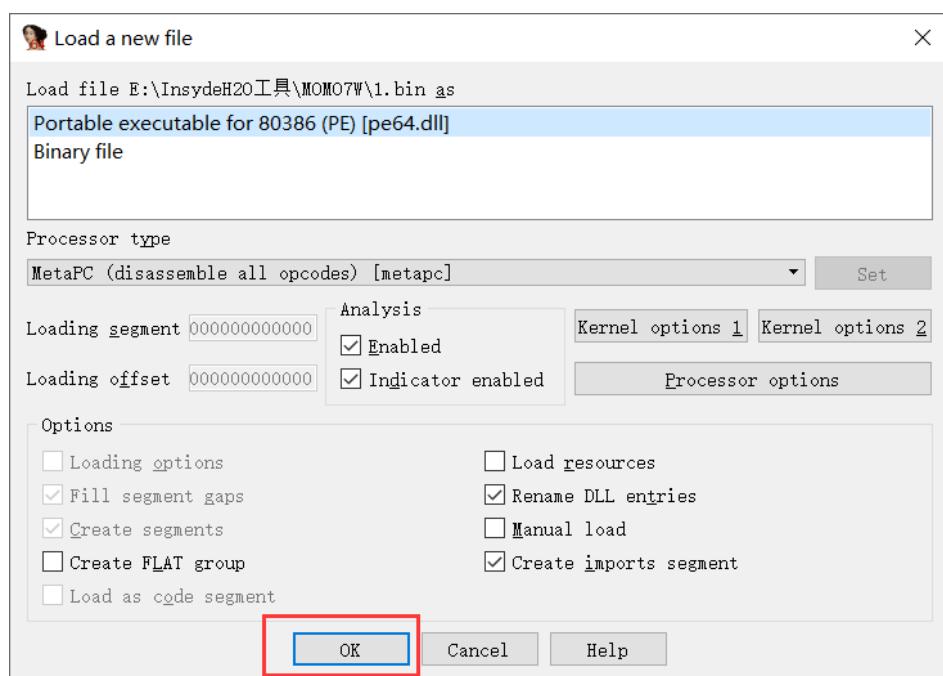
1. open IDA, click New after agreeing to a series of terms



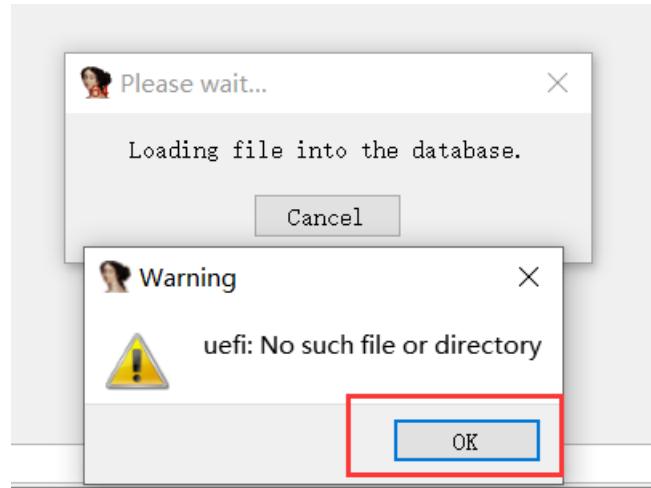
Open 1.bin file



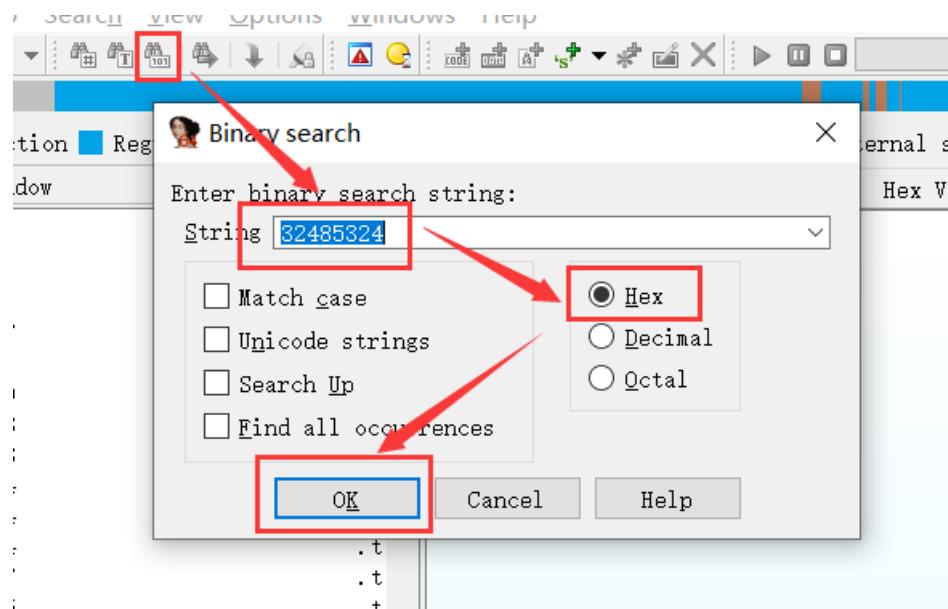
Click OK here directly



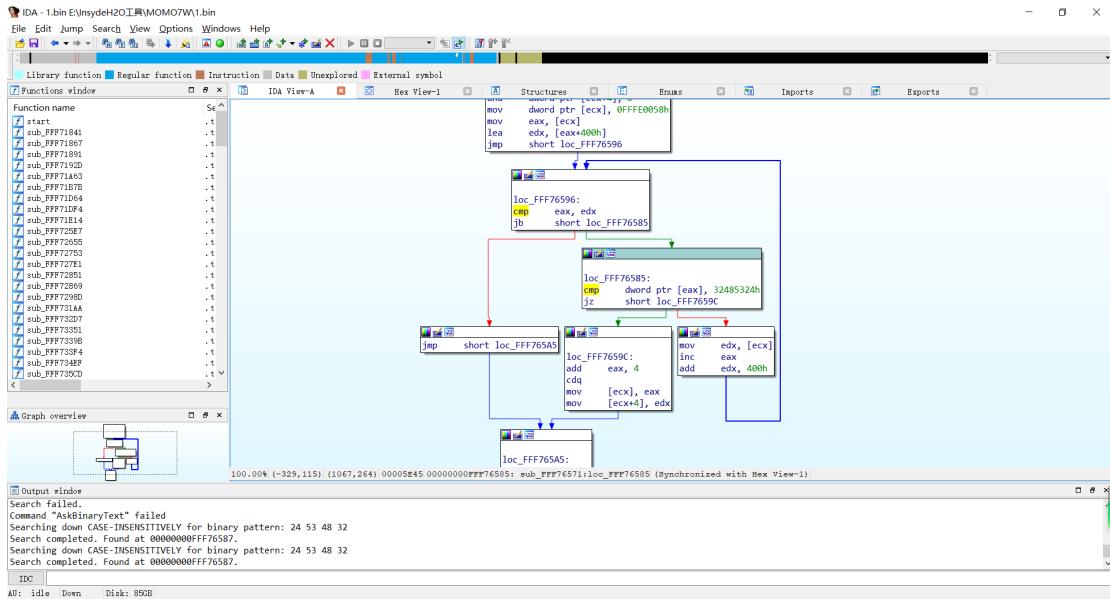
Because it is a free version, an warning message will pop up, please ignore the message and continue to click OK.



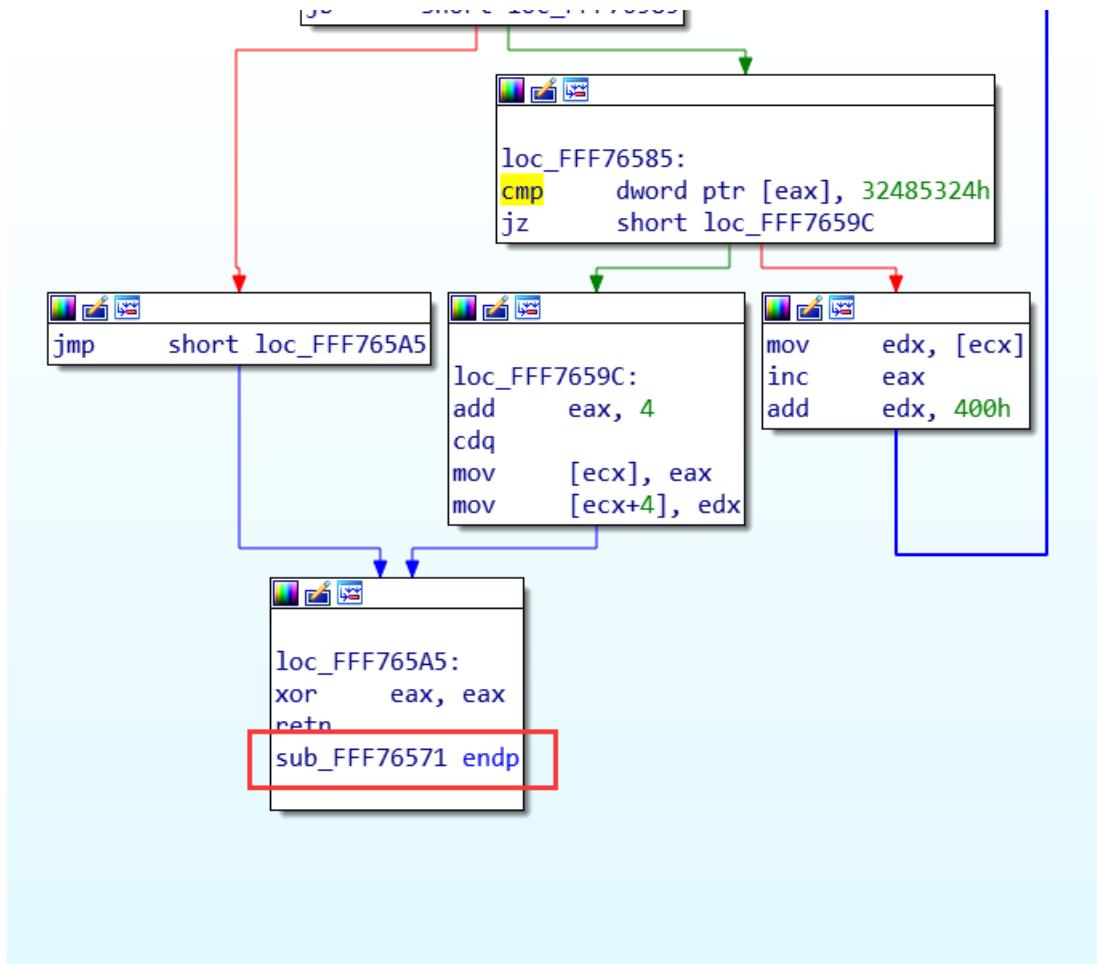
2. After loading, we will look for the identifier. Click Search for sequence of bytes on the toolbar, and enter 32485324 in the string box (this string of hexadecimal numbers is actually the reversed (2HS\$) hexadecimal ASCII code of \$SH2, because the order of data stored in memory is Little-Endian), then click OK to confirm.



3. After clicking OK, you will jump to this flow chart:



Let's go down and find the end of this subroutine



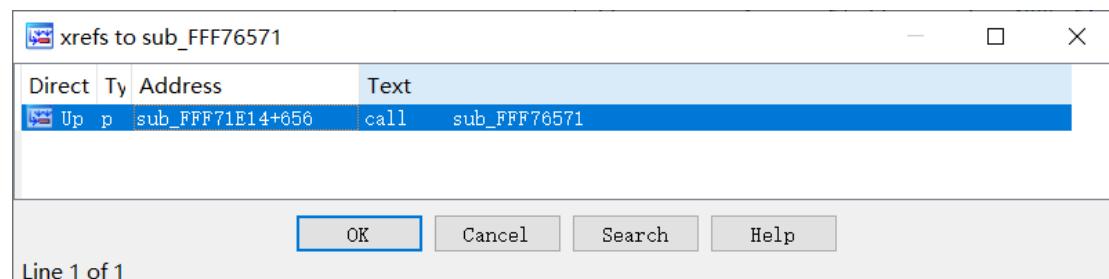
Click with the mouse to make it yellow

```

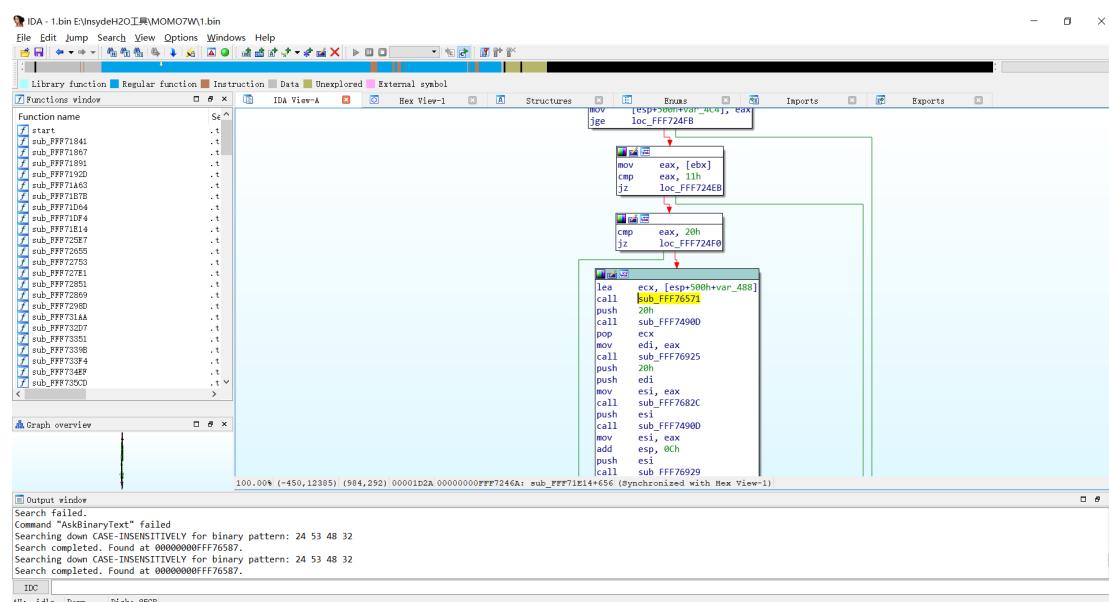
loc_FFF765A5:
xor    eax, eax
retn
sub_FFF76571 endp

```

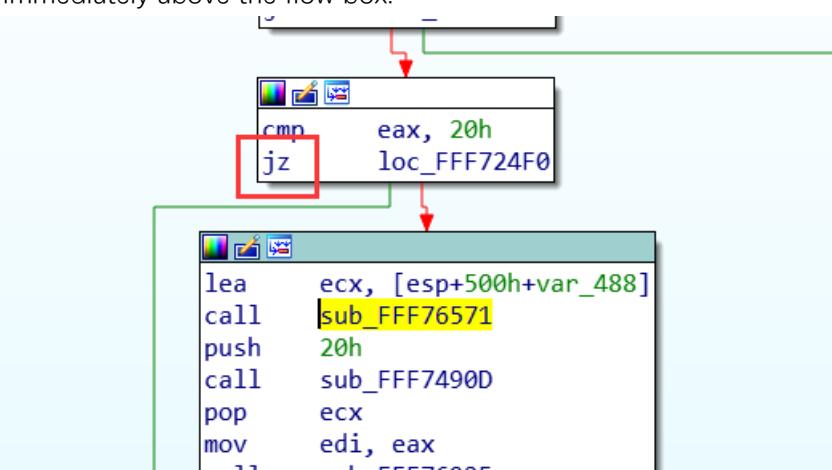
Then press the X key on the keyboard and click OK in the pop-up dialog



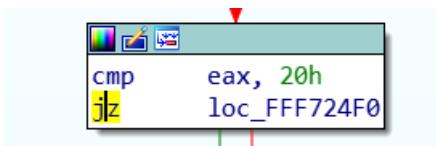
4. Next, we will see a flow chart similar to this:



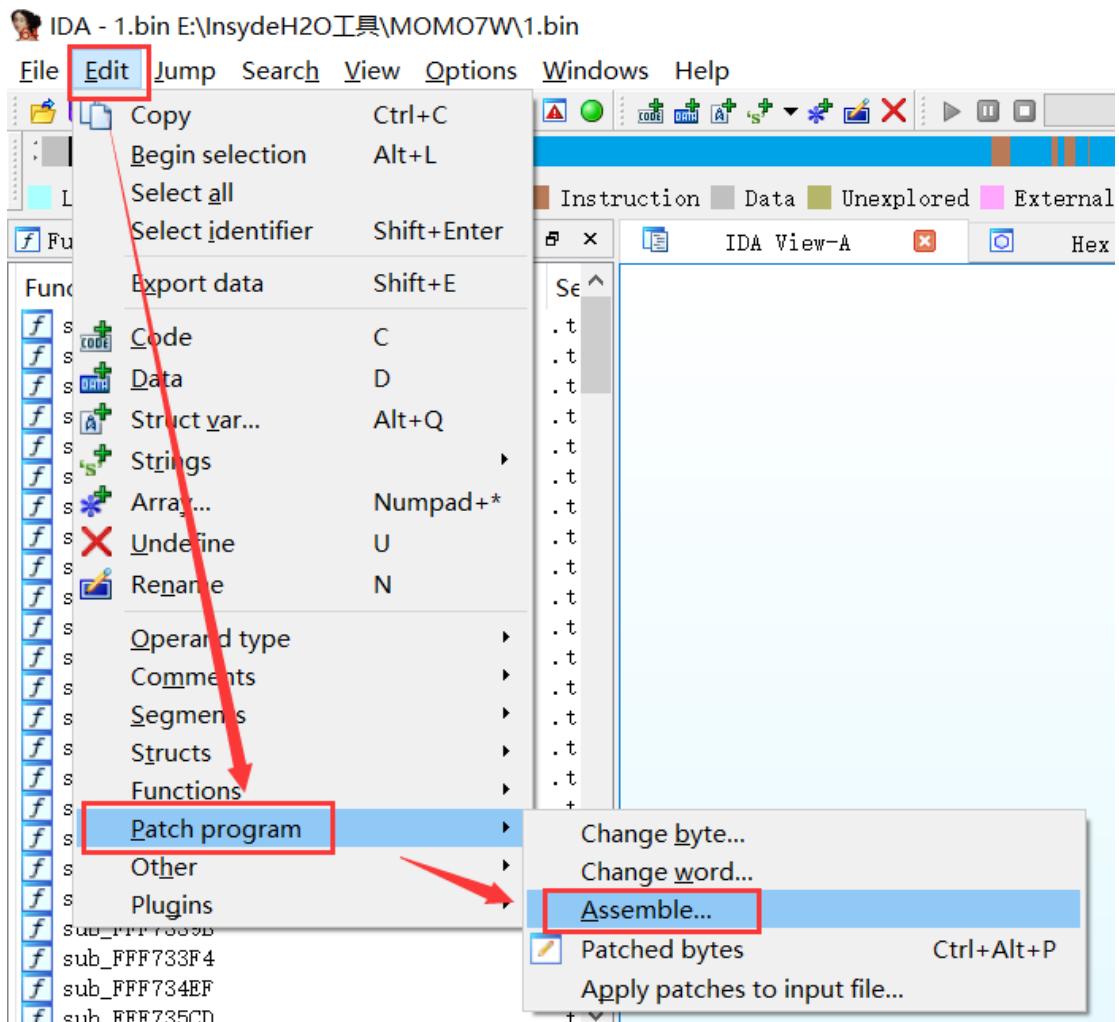
Click the jz immediately above the flow box:



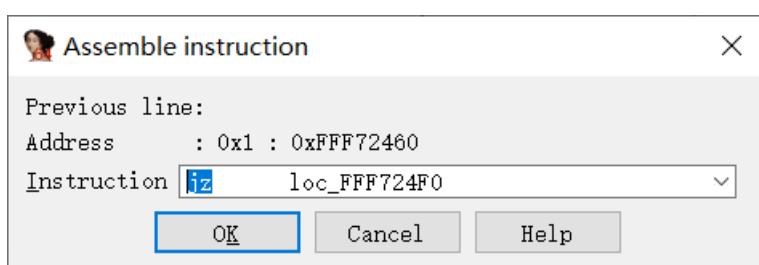
Becomes like this:



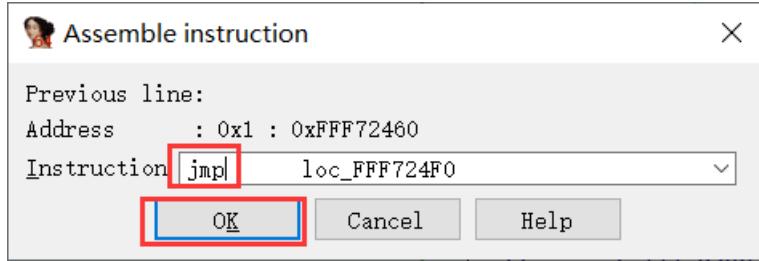
Then click the menu Edit->Patch program->Assemble



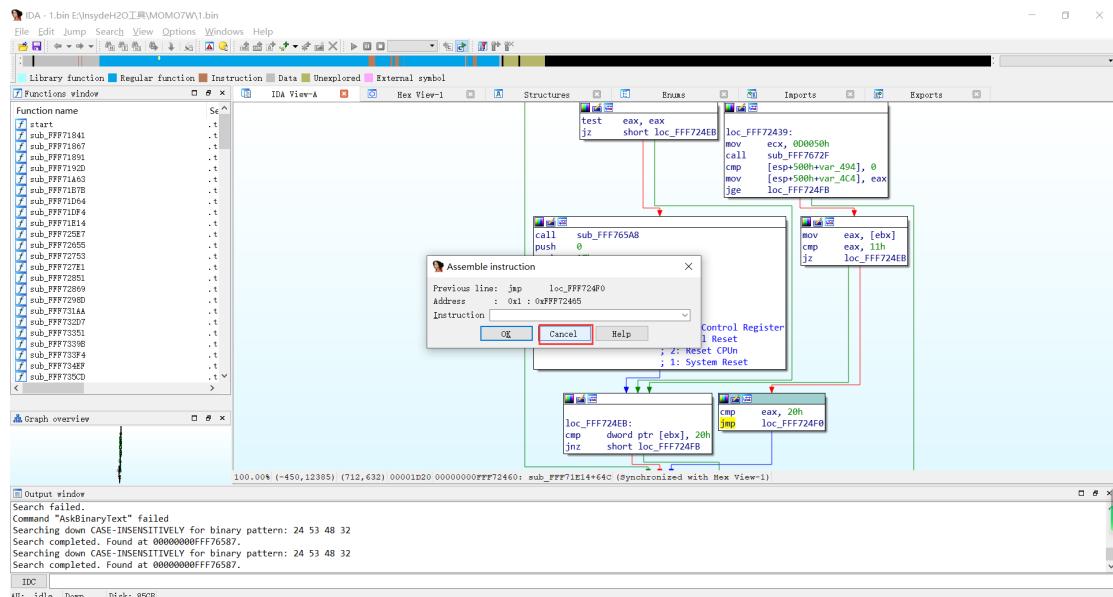
You will see this dialog:



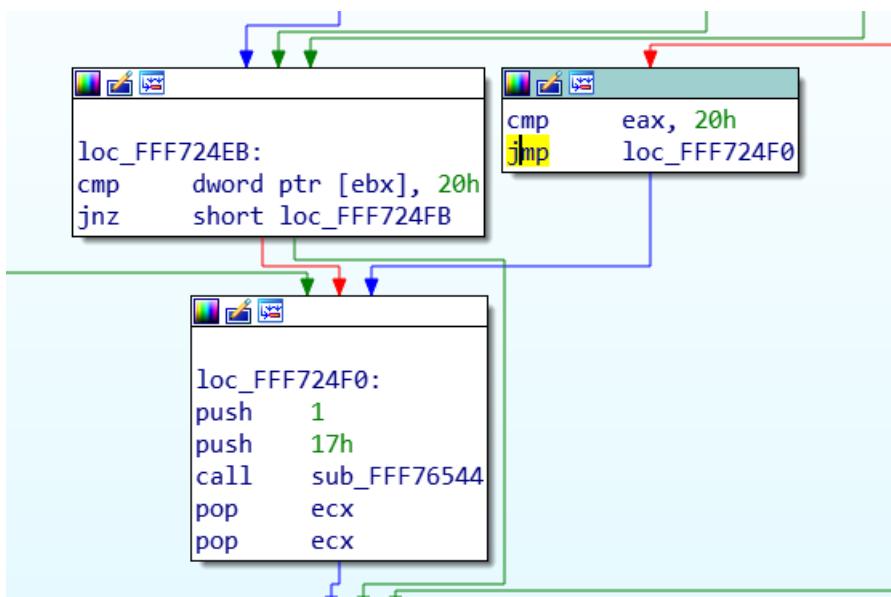
Directly change jz to jmp and click OK



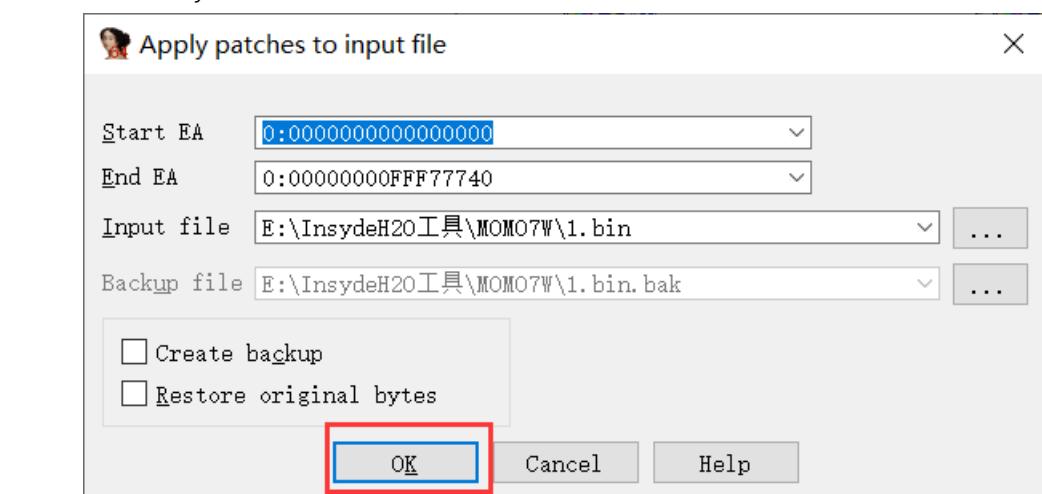
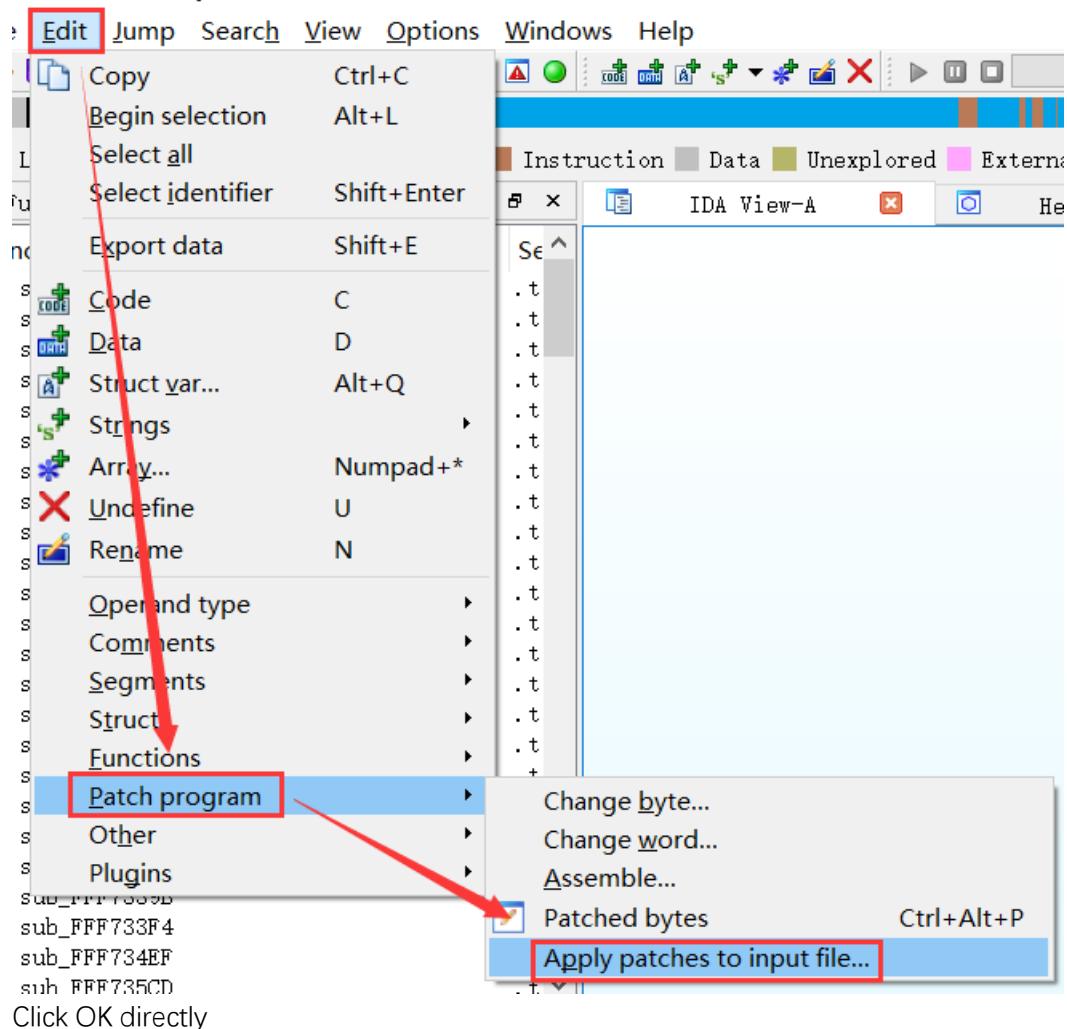
You will see a flow chart similar to this, click Cancel



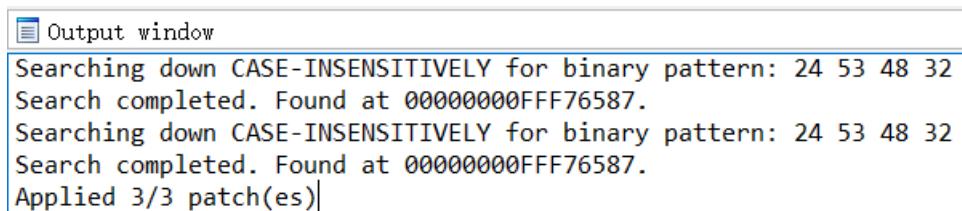
We successfully skipped the verification procedure



5. After the modification, the source file is actually not modified, and it needs to be applied manually. Click menu Edit-> Patch program-> Apply patches to input file



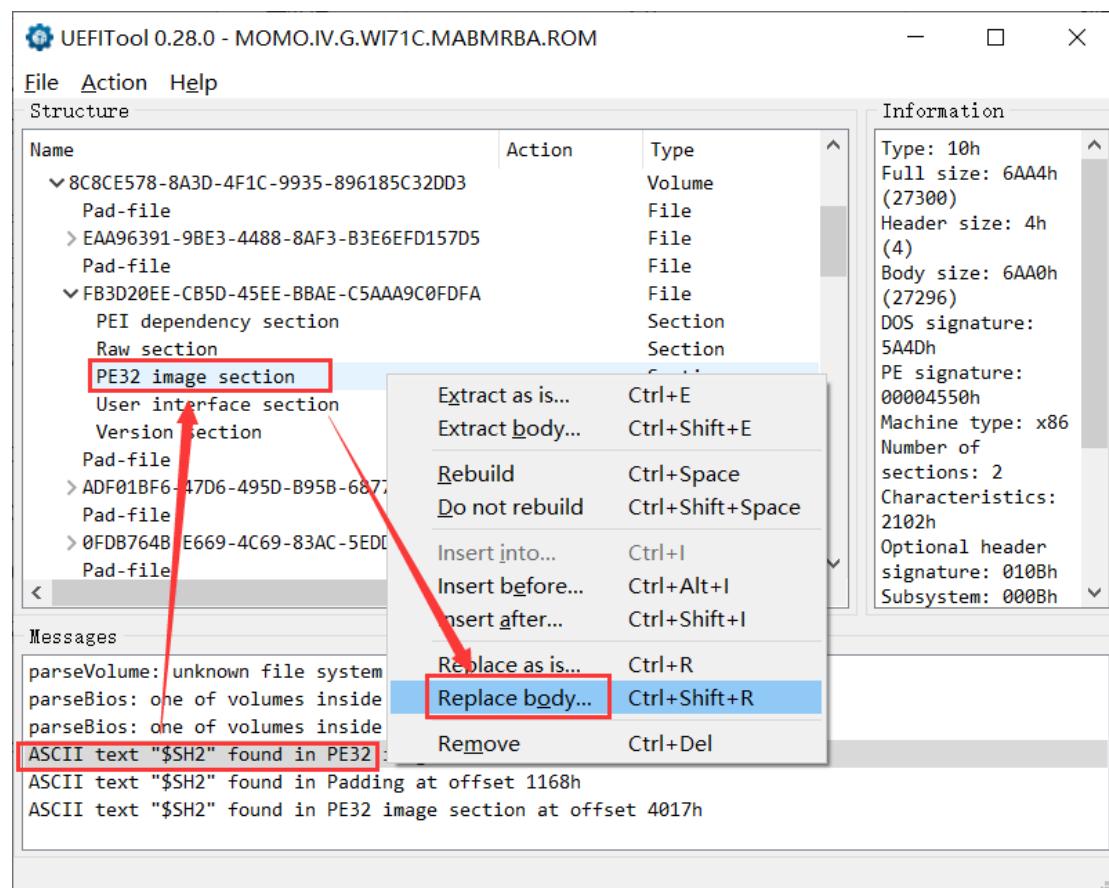
You will see an output like this below, indicating that the modification was successful.



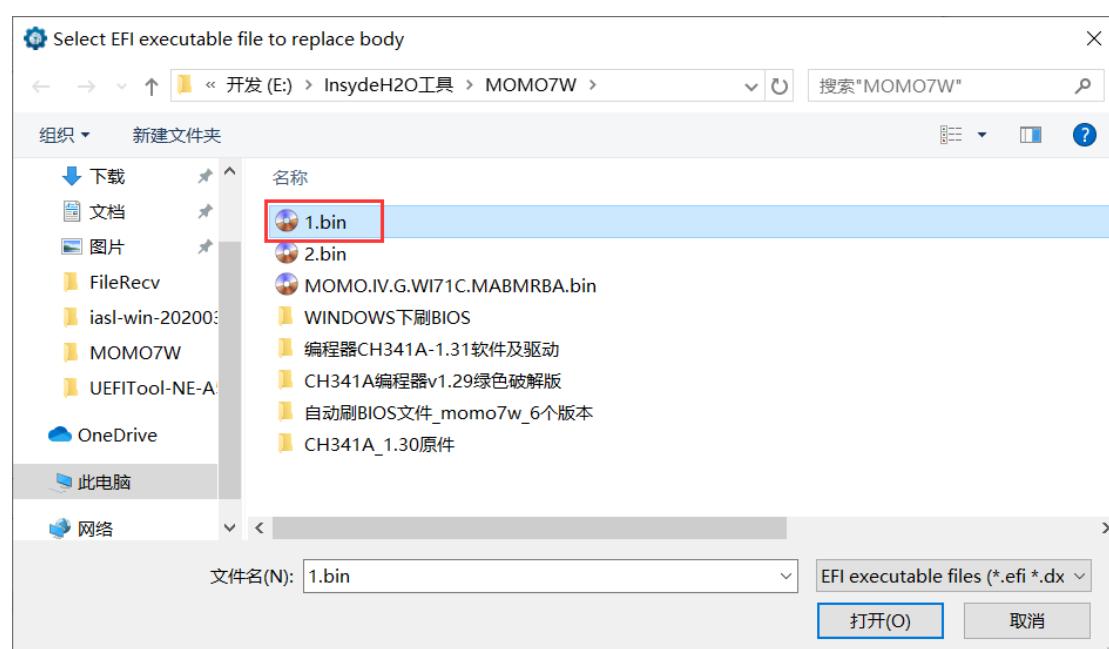
6. Because there are two verifications and the other is used as a backup program, the above operation needs to be repeated for 2.bin. Please do it yourself.

#### Step four. Pack UEFI (BIOS) firmware

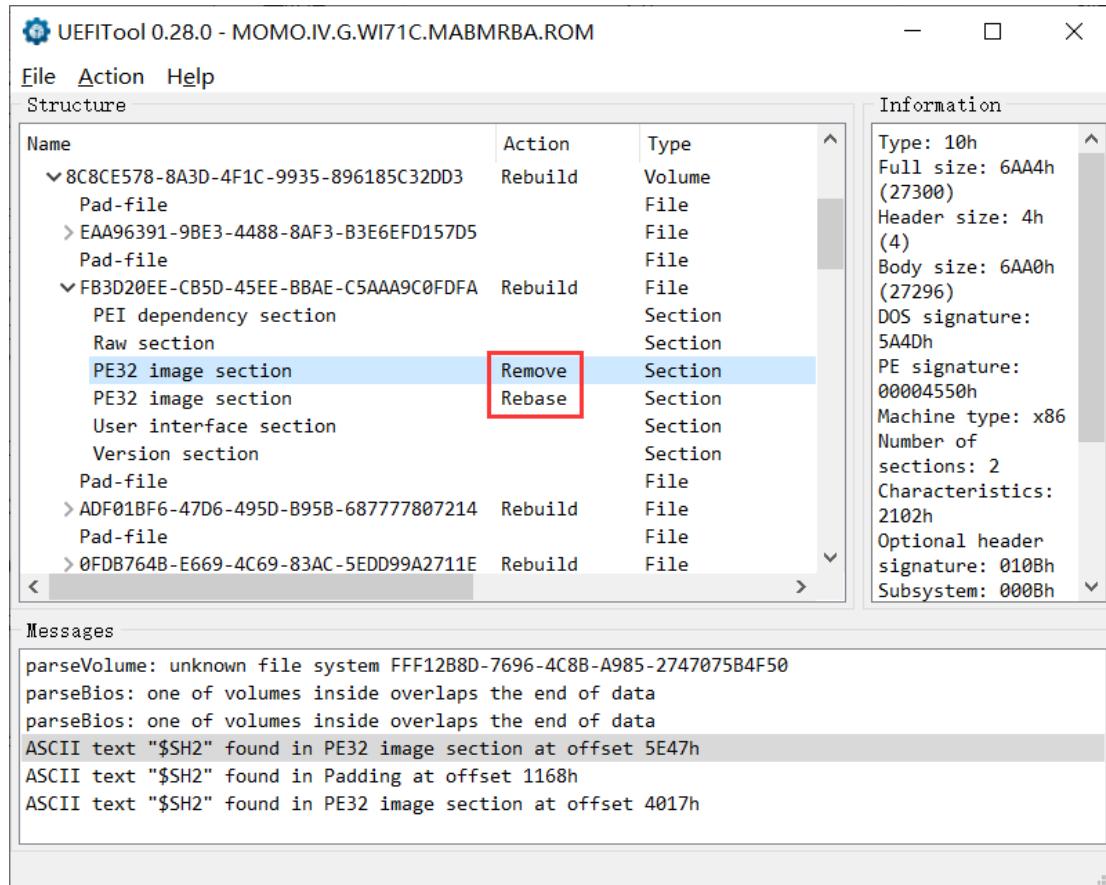
1. Go back to the UEFITool just now (if you accidentally turned it off, please reopen and load the BIOS file, and then search for \$SH2), double-click the first found PE32 image, right-click the PE32 image section, and click Replace body



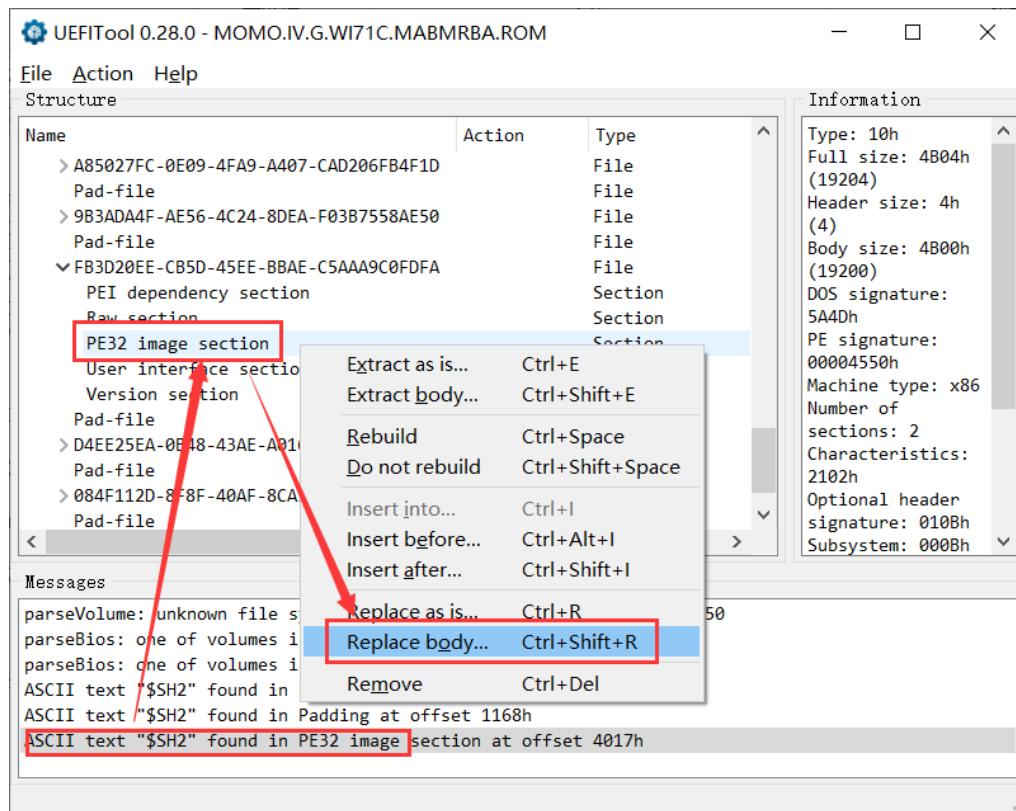
Open 1.bin

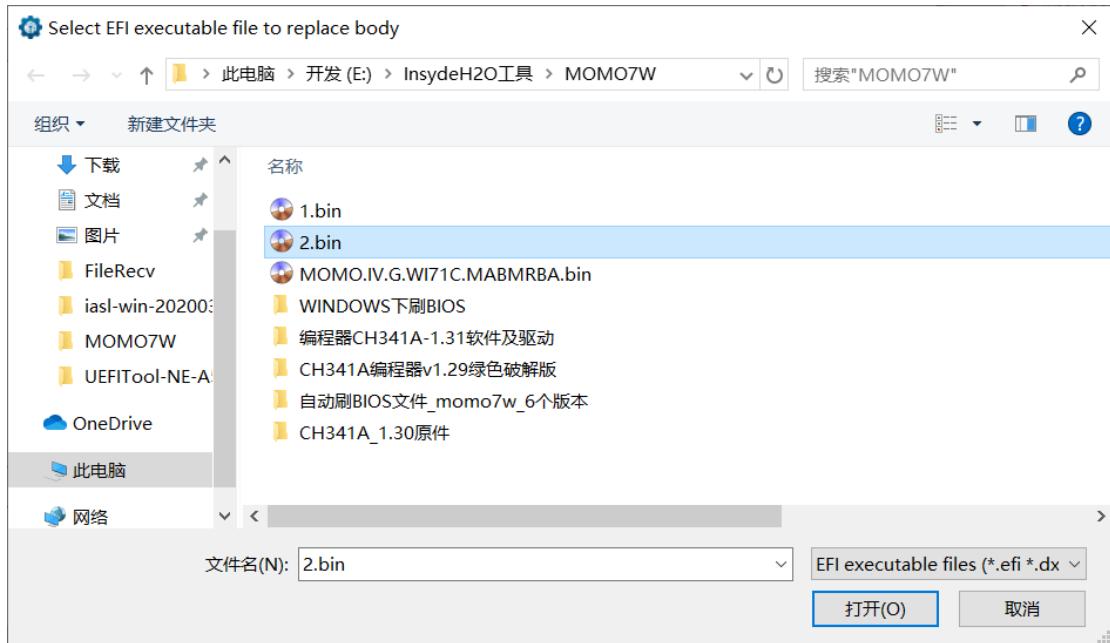


Then you will see a screen similar to this:



2. Repeat the previous operation, double-click the second found PE32, and then replace the body with the 2.bin file





**UEFITool 0.28.0 - MOMO.IV.G.WI71C.MABMRBA.ROM**

**File Action Help**

**Structure**

Name	Action	Type
> A85027FC-0E09-4FA9-A407-CAD206FB4F1D		File
Pad-file		File
> 9B3ADA4F-AE56-4C24-8DEA-F03B7558AE50		File
Pad-file		File
✓ FB3D20EE-CB5D-45EE-BBAE-C5AAA9C0FDFA	Rebuild	File
PEI dependency section		Section
Raw section		Section
PE32 image section	Remove Rebase	Section
PE32 image section		Section
User interface section		Section
Version section		Section
Pad-file		File
> D4EE25EA-0B48-43AE-A016-4D6E8B6C43B3	Rebuild	File
Pad-file		File
> 084F112D-8F8F-40AF-8CA5-1263B6403A51	Rebuild	File

**Information**

```

Type: 10h
Full size: 4B04h
(19204)
Header size: 4h
(4)
Body size: 4B00h
(19200)
DOS signature:
5A4Dh
PE signature:
00004550h
Machine type: x86
Number of
sections: 2
Characteristics:
2102h
Optional header
signature: 010Bh
Subsystem: 0008h

```

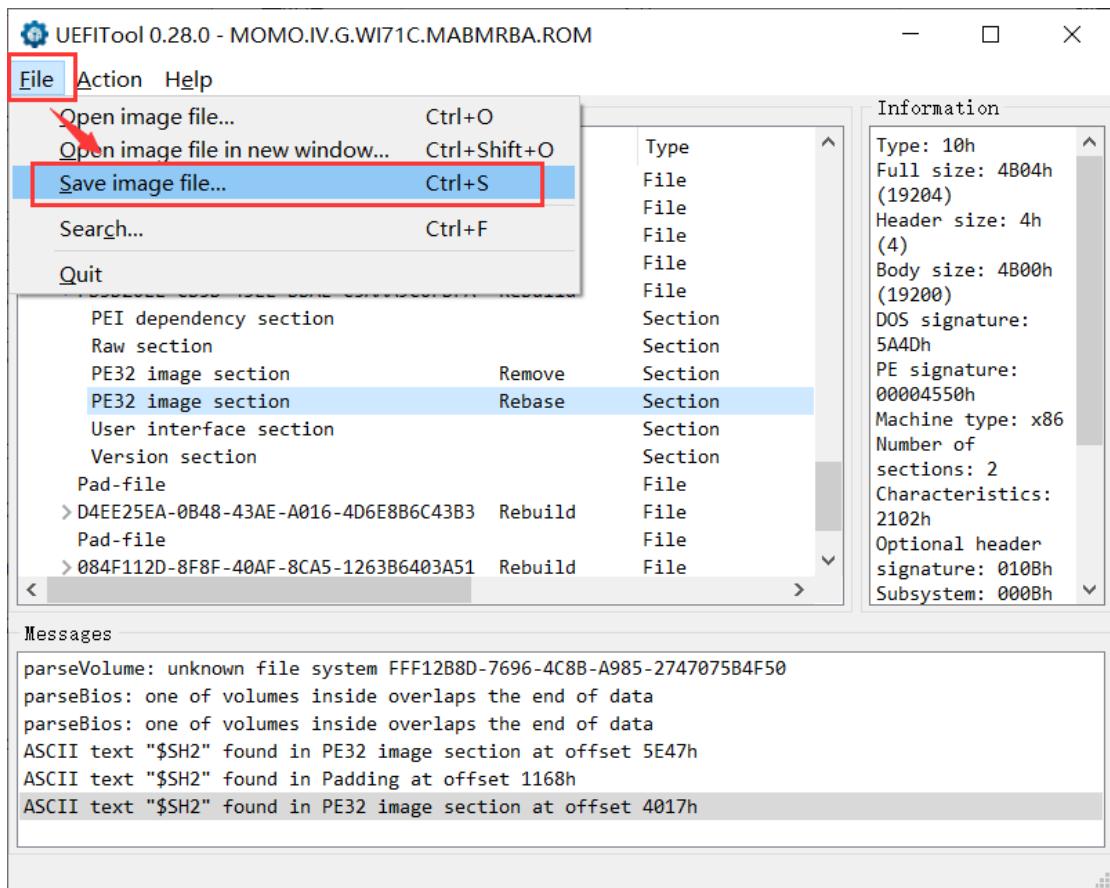
**Messages**

```

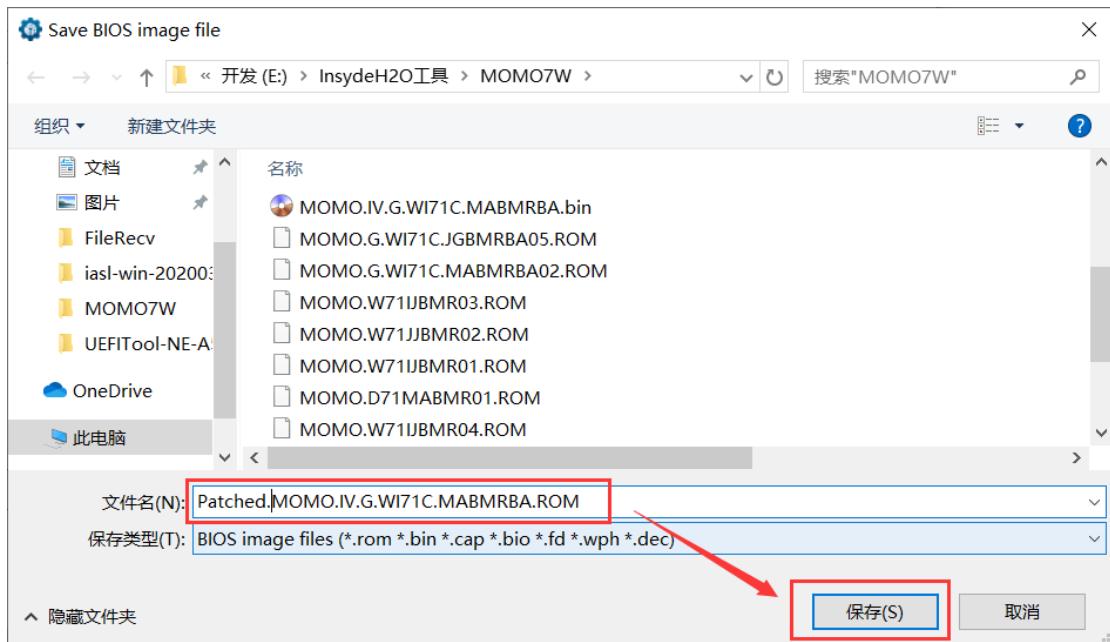
parseVolume: unknown file system FFF12B8D-7696-4C8B-A985-2747075B4F50
parseBios: one of volumes inside overlaps the end of data
parseBios: one of volumes inside overlaps the end of data
ASCII text "$SH2" found in PE32 image section at offset 5E47h
ASCII text "$SH2" found in Padding at offset 1168h
ASCII text "$SH2" found in PE32 image section at offset 4017h

```

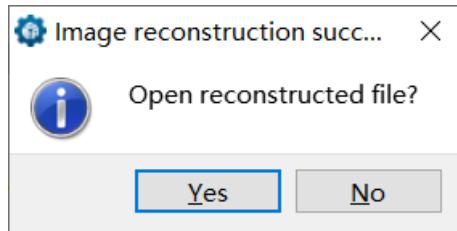
3. click the menu File-> Save image file to save the modified BIOS (UEFI) firmware



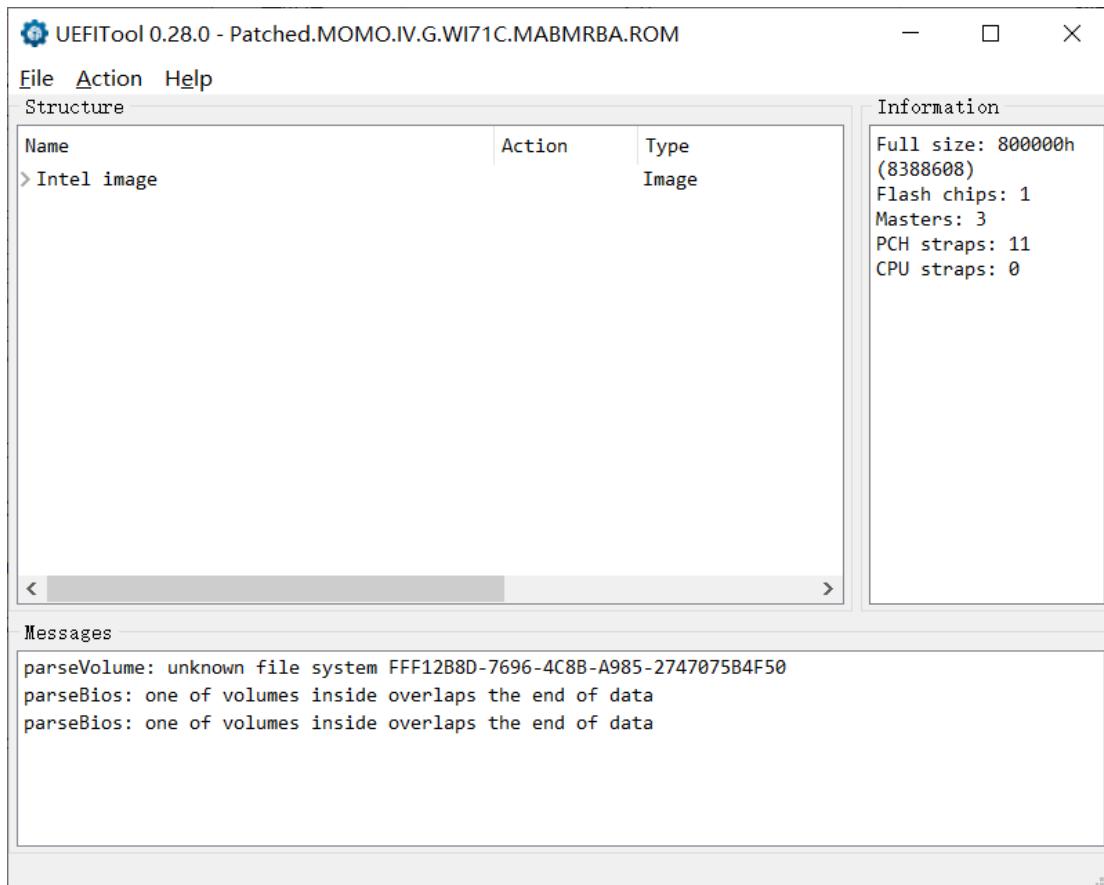
Rename it casually



Click Yes in the pop-up dialog



There is no new error in Messages box means it is modified successfully



### Step five. Flash BIOS

At this time, turn off UEFITool, use flash tools such as H2OFFT, FPTW to write BIOS into the chip, or flash it with a programmer. If the modified BIOS can boot successfully, it means the crack is successful. If it fails to boot, the modification fails, and you need to use the programmer to flash the original BIOS file again. The details can be found on the Internet.