Below is a completely worked‐out, step-by-step guide to extracting the flag from your PEM‐encoded RSA public key. We’ll show you:

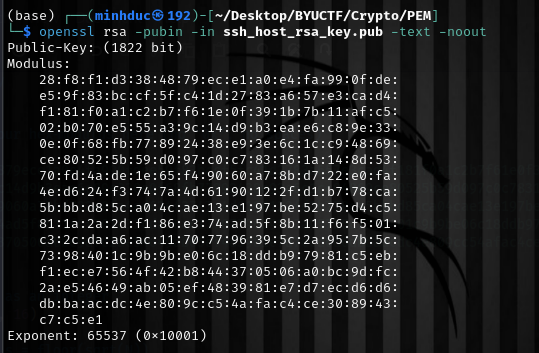
1. How to read the modulus N*N* out of the PEM file
2. Why “flag∈Nflag∈*N*​” tells you to take the integer square-root of N*N*
3. How to convert that big integer into bytes and recover the ASCII flag

**Why “flag ∈ √N” means √N is the flag**

* The statement  
  flag ∈ √N  
  is telling you that the **integer square‐root** of N*N* (i.e. ⌊N⌋⌊*N*​⌋) encodes your flag.
* Normally, N*N*​ in pure math is not an integer, but **floor**(N*N*​) is the unique integer s*s* with  
  s2≤N<(s+1)2.*s*2≤*N*<(*s*+1)2.
* That integer s*s* is large (about 1024 bits if N*N* is 2048 bits), but it *is* an exact integer. If you convert s*s* into a big‐endian byte‐string and then interpret that as ASCII text, you will recover your flag.

**Step 1: Extract N**

openssl rsa -pubin -in ssh\_host\_rsa\_key.pub -text -noout



**Step2: Dump N from the PEM:**

$ openssl rsa -pubin -in ssh\_host\_rsa\_key.pub -modulus -noout

modulus=28F8F1D3384879ECE1A0E4FA990FDEE59F83BCCF5FC41D2783A657E3CAD4F181F0A1C2B7F61E0F391B7B11AFC502B070E555A39C14D9B3EAE6C89E330E0F68FB77892438E93E6C1CC94869CE80525B59D097C0C783161A148D5370FD4ADE1E65F49060A78BD722E0FA4ED624F3747A4D6190122FD1B778CA5BBBD85CA04CAE13E197BE5275D4C5811A2A2DF186E374AD5F8B11F6F501C32CDAA6AC11707796395C2A957B5C7398401C9B9BE06C18DDB97981C5EBF1ECE7564F42B844370506A0BC9DFC2AE54649AB05EF483981E7D7ECD6D6DBBAACDC4E809CC54AFAC4CE308943C7C5E1

**Step3: Solve:**

