

Some computational problems in number theory

January 28, 2010

- (1) Say that I have a 2x2 matrix of the form:

$$A = \begin{bmatrix} 3 & * \\ * & * \end{bmatrix}$$

I don't know the matrix itself, but I do know that $\det(A) \equiv 17 \pmod{26}$ and I also know that

$$A \begin{bmatrix} 5 \\ 19 \end{bmatrix} \equiv \begin{bmatrix} 6 \\ 9 \end{bmatrix} \pmod{26}.$$

Find the matrix A .

- (2) In devising the RSA system you choose a public modulus $m = 1081 = 23 \cdot 47$ and an encrypting exponent of 73. Find the decrypting exponent.
(3) Calculate the Euler phi function of 864864. Use it to calculate

$$5^{207366} \pmod{864864}$$

- (4) (a) Compute $J(13, 4819)$
(b) Compute $13^{2409} \pmod{4819}$ (hint: $13^{29} \equiv 1 \pmod{4819}$)
(c) Is $13^{2409} \pmod{4819}$ prime? Why or why not?
(5) Determine if

$$x^2 + 12x + 10 \equiv 0 \pmod{53}$$

has a solution by completing the square and computing the appropriate Jacobi/Legendre symbol.

The next two problems require a computer

- (6) Find the next prime greater than or equal to

$$n = 10298374509348573904587390458732094587230495872309458723094573097$$

by testing if $a^{(n-1)/2} \equiv \pm 1 \pmod{n}$ for some values of a until you find a potential pseudo-prime and then convincing yourself that it is prime by checking if $J(a, n) = a^{(n-1)/2}$ for at least 10 values of a .

- (7) Say that your public modulus is:

$$\begin{aligned} m &= 9194050360213907115693366285304915215520274629853449561 \\ &= (9834710928479123480819)(934857203945872304958723049606019) \end{aligned}$$

and nobody else knows how your number factors. You also publish your public key to be:

$$3487192837645198273462939$$

which you choose at random so that it is relatively prime to $\phi(m)$. I send you the message 6001342142960307577337651863901327138891060326454897797, what does it say?

If you happen to be using Maple, I ran into trouble last week with the `ipowermod` function. I don't know what it is called. Here is the function that you can hopefully copy and paste.

```
ipowermod:=proc(a,b,n) local x;
if b<0 then return ipowermod(a,-b,n)^(-1) mod n;
elif b=0 then return 1;
elif b=1 then return a mod n;
elif b mod 2=0 then x:=ipowermod(a,b/2,n); return x^2 mod n;
else return a*ipowermod(a,b-1,n) mod n;
end;
end:
```