

MATH 470 Homework 4

1. Which of the following congruences have solutions

- a. $x^2 \equiv 11 \pmod{29}$
- b. $x^2 \equiv 23 \pmod{31}$
- c. $x^2 \equiv 5 \pmod{19}$
- d. $x^2 \equiv 7 \pmod{13}$

Since each modulus is prime, we know that $x \equiv a \pmod{p}$ will have a solution if and only if $a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$. Thus we have to compute $11^{14} \pmod{29}$, $23^{15} \pmod{31}$, $5^9 \pmod{19}$, $7^6 \pmod{13}$. Use the successive squaring algorithm for computing these (see web page). For example, the last of these is just $\equiv (49).(49).(49) \pmod{13}$. Since $49 \equiv 10 \pmod{13}$, this simplifies to $(10).(10).(10) \pmod{13}$ or $(-3).(-3).(-3) \pmod{13}$ or $-27 \pmod{13}$, that is -1 . Thus $x^2 \equiv 7 \pmod{13}$ has no solution.

As an alternative, and as you can see much faster method, we can compute the Legendre symbol $\left(\frac{a}{n}\right)$. For the first of these, we need $\left(\frac{11}{29}\right)$. Note both 11 and 29 are odd and in fact both are prime so that $\gcd(11, 29) = 1$. Then, $\left(\frac{11}{29}\right) = \left(\frac{29}{11}\right) = \left(\frac{7}{11}\right) = -\left(\frac{11}{7}\right) = -\left(\frac{4}{7}\right) = -\left(\frac{2}{7}\right)^2 = -1$. Thus we can conclude that 11 is not a square root $\pmod{29}$. For the second problem we have: $\left(\frac{23}{31}\right) = -\left(\frac{31}{23}\right) = -\left(\frac{8}{23}\right) = -\left(\frac{2}{23}\right)^3$. We can compute $\left(\frac{2}{23}\right)$ directly from the theorem: since $23 \equiv 7 \pmod{8}$ it is $+1$ and so $\left(\frac{23}{31}\right) = -1$. This shows there is no solution to this congruence.

Finally, we compute $\left(\frac{5}{19}\right)$: $\left(\frac{5}{19}\right) = \left(\frac{19}{5}\right) = \left(\frac{4}{5}\right) = \left(\frac{2}{5}\right)^2 = 1$. Thus $5^9 \pmod{19}$ does have a solution. We can assert this since 19 is prime (but could not if it were composite).

2. For those above congruences that have solutions, can you find them?

Here $19 \equiv 3 \pmod{4}$ so we know that a solution is $\pm x$ where $x \equiv 5^5 \pmod{19}$. We simplify this successively to $(25).(25).(5) \pmod{19}$, $(6).(6).(5) \pmod{19}$, $(-2).(5) \pmod{19}$, $9 \pmod{19}$. Indeed, $9^2 \equiv 81 \equiv 5 \pmod{19}$.

3. Find the solutions(s) of the congruence $x^2 \equiv 43 \pmod{91}$

Here 91 is composite with factors 7 and 13. So $x^2 \equiv 43 \pmod{91}$ also means $x^2 \equiv 1 \pmod{7}$ and $x^2 \equiv 4 \pmod{13}$. Can we solve these? Quite clearly, yes. For the first since $\pm 1^2 \equiv 1 \pmod{7}$ and for the second $2^2 \equiv 4 \pmod{13}$. We have to combine these four solutions by the Chinese Remainder Theorem. This means finding integers s and t such that $7s + 13t = 1$. The extended Euclidean algorithm provides $s = 2$ and $t = -1$. [Alternatively given the small numbers involved, we can just solve $7s \equiv 1 \pmod{13}$ and $13t \equiv 4 \pmod{7}$ by trying

small values.] Now the answer is just $x = (7).(2).(\pm 2) + (13).(-1).(\pm 1) \pmod{91}$. Taking both the positive possibilities these expand to $x \equiv 15 \pmod{91}$, while the positive for the first and negative for the second gives $x \equiv 41 \pmod{91}$. We also have the negative of these; $x \equiv -15 \equiv 76 \pmod{91}$ and $x \equiv -41 \equiv 50 \pmod{91}$. Thus $x = \{15, 41, 50, 76\} \pmod{91}$.

4. Alice and Bob secretly each week at one of six locations - which they have labelled 1-6. When one wants to meet the other they send a message with the place, date, time. Specifically, a message is a string of 4 digits: the first digit gives the place; the second gives the day of the week (with values [Sunday=0, ... Saturday=6]); the final two the time on the hour (with values [0-23]). Thus to meet next at location 4 on Monday (day = 1) at 9pm (= 21 hour), the message would be $m = 4121$. If it were next Saturday at 7am Alice would want to send Bob the message $m = 4607$. The encrypted message is sent over a public network using Rabin's method. Bob's public key is $n = 3545233$. Alice sends Bob c where $m^2 \equiv c \pmod{n}$. When Bob receives the codetext c he decrypts by solving $x^2 = c \pmod{n}$. But Bob knows the factors of n (which are $p = 1627$, $q = 2179$). and so uses this to recover $x = m$. If he receives the codetext 380062 can Bob decipher the message uniquely, and if so, when and where should he meet Alice? Roughly, what are the chances of being able to do this, that is, in finding a unique message of the correct form? Think about this carefully!

For the solution, note that both p , q are $\equiv 3 \pmod{4}$ so that square roots mod p and q can be obtained by formula. Then combine with the Chinese Remainder Theorem. The web page also provides a *Maple* worksheet for this problem.

For the next parts; first, what will the codetext look like? It will be a number in the range $[0, n - 1]$ and we should certainly **not** expect all of these numbers to be equally probable from even purely randomly generated plaintext consisting of numbers in the 10's of digits. But it would be surprising if equal length, large block sizes of this interval didn't have roughly equal distribution of codetext values. When we run the square root computations and re-combine these with the Chinese remainder theorem we should expect that the distribution of possible messages m are also approximately equally distributed. If this is the case then we should roughly expect ten times the number of two digit messages over single digits ones. We should also expect the number of 3 digit messages should be tenfold that of the two digit ones. Continuing this argument, we should expect 10 times the number of 5 digit messages and 100 times the number of 6 digits ones than those with 4 digits, etc. So all together, there are 9000 possible 4-digit values of computed m and $n \approx 3,500,000$ total possible values of m . The chance of a random solution being 4 digits long is therefore $9000/3,500,000$ or about $1/4000$. Since we have 4 values of m generated, one is certainly 4 digits long, but the other 3 will have less than one chance in a thousand of being 4 digits. For the example given, not all 4 digit computed values of m are possible plaintexts - in fact only about $\frac{1}{10}$ will be. Thus overall, the probability of getting a second feasible plaintext is roughly $1 : 10^{-4}$.

I ran the cipher with all possible values of the plaintext message and here is what happened. With the given value of n 12 "potential solutions" had x with 2 or less digits; 14 had exactly 3 digits; 13 had exactly 4 digits; 123 had exactly 5 digits; 1476 had exactly 6 digits; 3086 had exactly 7 digits. When I used a different n ($p = 1987$, $q = 2311$) I got the values: 12, 15, 8, 102, 1219, 3368. So, over large intervals, the spread is indeed roughly proportional to the values in $[0, n - 1]$, but

with a tendency to bunch near very small numbers.

To improve the chances of an unambiguous message Alice and Bob agree to modify the message to one of length 5 digits by simply padding the last digit to be a 0 (so that valid messages will always end in 0). Does this help and by how much?

It has no effect whatsoever!. The above argument explains why. If we did as suggested the number of plausible, computed messages would be again just 9,000. The extra digit is fixed at zero and so contributes nothing.

A few further observations are in order.

First the size of the primes chosen here is many orders of magnitude too small for anything other than illustration. Each of p and q are typically in the 100 digit range. It is only in this range that factoring is (currently) computationally infeasible.

Second, the length of the plaintext would be a problem regardless of how big we had chosen n . There are only about 1,000 possible plaintexts. Eve would simply try each one in turn and compare the ciphertext to the one intercepted. The cost: 1,000 encrypts - computationally each is just one square and taking the remainder $(\text{mod } n)$. Even a billion of these would be very fast on a desktop machine..

Finally, if anyone were given the set of computed solutions then the factorisation of n would be straightforward. Suppose we have x_1 and x_2 (one could be the message, but the other must be different). If $x_1^2 \equiv c \pmod{n}$ and $x_2^2 \equiv c \pmod{n}$, then $x_1^2 - x_2^2 \equiv 0 \pmod{n}$. This means that $x_1 \pm x_2$ must be divisible by either p or q . We then take $\text{gcd}(x_1 + x_2, n)$ to obtain p (or q) and we have factored n . Note the last computation is just the Euclidean algorithm and is extremely fast. In the present case, the four solutions computed are: $\{x_1, x_2, x_3, x_4\} = \{4619, 121763, 3423470, 3540614\}$. $\text{gcd}(x_1 - x_3, n) = 2179 = q$ and $\text{gcd}(x_3 - x_4, n) = 1627 = p$.

5. Bob Blunt's public Rabin key is the number $n = 1219337183957664455746356446574002227$. His published protocol is that text letters should simply written as their two digit decimal equivalents with space the value 00 so that : $a = 01, b = 02, \dots z = 26$. Sally Silly sends Bob a message encrypted using this system and the cipher text is $c = 654481$. This is intercepted by Eve who finds the cipher text puzzling. She is not at home and doesn't have access to her advanced cryptographic tools, only a simple calculator. However, after a little thought, she quickly decrypts it. How did she do this and what is the message?

So there is a legitimate message m , squared and then taken $(\text{mod } n)$ to obtain a value c . If c is truly random in the range $(0, n)$ and n has k digits then 90% of the values of c should have k digits, 90% of the rest should have $k - 1$ digits, etc. Here n has 38 digits yet c has only 6. Chances of this happening for a random value of c is about one in 10^{-32} . Unlikely. So this is indeed puzzling. Thus c is surely not random but comes from a very specialized m - one that only manages a 1-in- 10^{-32} probability. Most likely reason is m is small and m^2 is still small in relation to n . But in this case $m^2 < n$ and so there is no modular arithmetic involved. To find m we just take the (regular) square root of c to recover the plaintext.

Moral of story; must pad the plaintext m to avoid this issue!