Math 259 - Spring 2019
Homework 1 Solutions

1. This problem is just to make sure that everyone can do some basic computations using any software they are comfortable with. I am personally using Sage, but anything will do.

   (a) We have that $N = 27894437$, and $\varphi(N) = (p-1)(q-1) = 3700 \times 7536 = 27883200$. Then we have that $d \equiv e^{-1} \equiv 443^{-1} \equiv 10259507 \pmod{27883200}$.

   (b) To encrypt we simply compute $c \equiv m^e \equiv 11034007^{443} \equiv 19717832 \pmod{27894437}$.

   (c) To decrypt we compute $m \equiv c^d \equiv 3003890^{10259507} \equiv 12990712 \pmod{27894437}$.

2. To get $N = pq$ from $\varphi(N) = (p-1)(q-1)$, we need a relationship between N and $\varphi(N)$. Expanding $\varphi(N)$, we have that $\varphi(N) = (p-1)(q-1) = pq - p - q + 1 = N - p - q + 1$. We can solve this for $p + q$ and say that

$$p + q = N - \varphi(N) + 1 = 3259499 - 3255840 + 1 = 3660.$$

   So I'm looking for two numbers p and q such that $pq = 3259499$ and $p + q = 3660$. This is two equations in two unknowns, which I should be able to solve. I can say that $p = 3360 - q$, and substituting into the first equation I get that

$$3259499 = pq = (3660 - q)q = 3660q - q^2,$$

   or

$$q^2 - 3360q + 3259499 = 0.$$

   This can be solved using the quadratic formula:

$$q = \frac{3660 \pm \sqrt{3360^2 - 4 \times 3259499}}{2} = \frac{3660 \pm \sqrt{357604}}{2} = \frac{3660 \pm 598}{2} = 1531 \text{ or } 2129.$$

   Turns out that q can be either, and then p will be the other one (check this using the relation $p = 3660 - q$). So $N = 1531 \times 2129$.

3. Just so we don't have to keep writing such big numbers, let

$$x = 516107, \quad y = 187722, \quad \text{and} \quad N = 642401.$$

   Then putting the two given congruences together, we get that

$$x^2y^2 = (xy)^2 \equiv 2^2 \cdot 7^2 = 14^2 \pmod{N}.$$

   Another way to write this is as

$$(xy)^2 - 14^2 \equiv 0 \pmod{N},$$

or

$$(xy - 14)(xy + 14) \equiv 0 \pmod{N}.$$

To be explicit, we have that

$$xy - 14 = 96884638240$$

and

$$xy + 14 = 96884638268.$$

How does any of this help us? I'm glad you ask. We have that

$$96884638240 \times 96884638268 \equiv 0 \pmod{N},$$

and $N = pq$ for some product of two primes. This means that there is an integer k such that

$$96884638240 \times 96884638268 = kN = kpq.$$

Now since p and q are primes, they don't "break up" any more under multiplication. So it is forced that p divides either 96884638240 or 96884638268, and same for q. In other words, we must have that

$$\gcd(96884638240, N) > 1 \quad \text{or} \quad \gcd(96884638268, N) > 1.$$

Now we may just hope for the best (that we don't have $\gcd(96884638240, N) = 1$ and $\gcd(96884638268, N) = N$ or vice-versa, but graduate students will prove that this doesn't happen ever!) and compute the gcd:

$$\gcd(96884638240, N) = 1129,$$

and

$$\gcd(96884638268, N) = 569,$$

which in fact does factor N.

4. First, we have that for each $i = 1, 2, \ldots, k$, we have

$$c_i \equiv m^e \pmod{N_i}.$$

Therefore, we also have

$$c \equiv m^e \pmod{N_i},$$

for each $i = 1, 2, \ldots, k$.

Now since this one number works modulo each N_i, it must also work modulo the product of the N_is, i.e.

$$c \equiv m^e \pmod{\prod_{i=1}^{k} N_i}.$$

This is ensured by the Chinese Remainder Theorem, which states that there is a unique simultaneous "lift" of classes modulo N_i for each i to a class modulo $\prod N_i$.

So far nothing very special has happened. Now comes the magic: Since $m < N_i$ for each i, and $e \leq k$, we must have that

$$m^e < \prod_{i=1}^{k} N_i.$$

This is because on the left there are few small numbers multiplied together and on the right there are many big numbers multiplied together.

We also have that $c < \prod_{i=1}^{k} N_i$. But two numbers that are less than $\prod_{i=1}^{k} N_i$ and equal modulo $\prod_{i=1}^{k} N_i$, must be actually equal **as integers!**

Therefore

$$c = m^e$$

full stop, no congruence. And $m = \sqrt[e]{c}$. Now this is a root in the integers (not modulo anything) which is easy to compute.

5. This time we have two ciphertexts, c_1 and c_2, and we have

$$c_1 \equiv m^e \pmod{N} \quad \text{and} \quad c_2 \equiv m^f \pmod{N},$$

with the same N and m.

Using the hint, we assume that Eve knows a and b with $ae + bf = 1$. Then Eve wins by computing $c_1^a c_2^b \pmod{N}$, because we have that

$$\begin{aligned} c_1^a c_2^b &\equiv (m^e)^a (m^f)^b \pmod{N} \\ &\equiv m^{ae} m^{bf} \pmod{N} \\ &\equiv m^{ae+bf} \pmod{N} \\ &\equiv m \pmod{N}. \end{aligned}$$

6. Bob will send

$$c_1 \equiv g^b \equiv 5^{33} \equiv 7 \pmod{73}$$

and

$$c_2 \equiv m \cdot h^b \equiv 62 \cdot 49^{33} \equiv 68 \pmod{73}.$$

7. I found $a = 156$ by brute force. It was fast because the numbers are relatively small, but there is nothing really smart I can think to do. However, if you know enough Python it's not too annoying:

```
for i in range(1223):
    k = 5**i  % 1223
    if k == 3:
       print i
```

8. (a) $\log_3 1 = 0$
   (b) $\log_3 3 = 1$
   (c) $\log_3 5 \equiv \log_3(7 \times 8) \equiv \log_3 7 + \log_3 8 \equiv 11 + 10 \equiv 21 \equiv 5 \pmod{16}$
   (d) Since $1 \equiv 10 \times 12 \pmod{17}$, we have that $\log_3 1 \equiv \log_3 10 + \log_3 12 \pmod{16}$ or $0 \equiv 13 + \log_3 10 \pmod{16}$. Then $\log_3 10 \equiv -13 \equiv 3 \pmod{16}$.

9. (a) We have that $2^7 \equiv 3^3 \pmod{101}$. Taking $\log_3$ on each side, this gives the equation

$$\log_3(2^7) \equiv \log_3(3^3) \pmod{100}$$

   which we can simplify:

$$\begin{aligned} 7\log_3 2 &\equiv 3\log_3 3 \pmod{100} \\ 7\log_3 2 &\equiv 3 \cdot 1 \pmod{100} \\ 7\log_3 2 &\equiv 3 \pmod{100}, \end{aligned}$$

   which we can solve to say that $\log_3 2 \equiv 3 \cdot 7^{-1} \equiv 29 \pmod{100}$.
   (b) We have $b = 6$.
   (c) We do the same trick as in part (d) of problem 8: Since $17 \times 6 \equiv 1 \pmod{101}$, we have that

$$\begin{aligned} \log_3 17 + \log_3 6 &\equiv \log_3 1 \pmod{100} \\ \log_3 17 + \log_3 6 &\equiv 0 \pmod{100}, \end{aligned}$$

   so $\log_3 17 \equiv -\log_3 6 \pmod{100}$. Now we notice that $6 = 2 \times 3$, so we can break this up further:

$$\log_3 17 \equiv -(\log_3 2 + \log_3 3) \equiv -\log_3 2 - 1 \pmod{100}.$$

   From part (a), $\log_3 2 \equiv 29 \pmod{100}$, so

$$\log_3 17 \equiv -29 - 1 \equiv -30 \equiv 70 \pmod{100}.$$

Extra problems for graduate credit:

1. (a) Say that $k = \ell\varphi(N)$. We have that $\varphi(N) = (p-1)(q-1)$, so

$$a^k \equiv a^{\ell(p-1)(q-1)} \equiv (a^{p-1})^{\ell(q-1)} \equiv 1^{\ell(q-1)} \equiv 1 \pmod{p},$$

   where here we used Fermat's Little Theorem, which we can apply because $\gcd(a, N) = 1$ implies that $\gcd(a, p) = 1$. Similarly for q in place of p.

(b) Again the same argument will apply with q in place of p so we only show that $a^{k+1} \equiv a \pmod p$. If $\gcd(a,p) = 1$, by part (a) we are done by simply multiplying both sides of the congruence by a.

If $\gcd(a,p) \neq 1$, then it must be the case that $\gcd(a,p) = p$, since p is prime and its only divisors are 1 and p. In particular, this means that p divides a or $a \equiv 0 \pmod p$. Therefore $a^{k+1} \equiv 0 \pmod p$ and $a \equiv 0 \pmod p$, from which it follows that $a^{k+1} \equiv a \pmod p$.

(c) From part (b), since $a^{k+1} \equiv a \pmod p$ and $a^{k+1} \equiv a \pmod q$, by the Chinese Remainder Theorem it follows that $a^{k+1} \equiv a \pmod N$ as well, for arbitrary a and arbitrary k a multiple of $\varphi(N)$.

Now if e and d are encryption and decryption exponents for RSA with modulus N, this means that $ed \equiv 1 \pmod{\varphi(N)}$, or that there is an integer k which is a multiple of $\varphi(N)$ with $ed = 1 + k$. Now the result follows.

2. Note that actually for this problem to be correct, p and q must be odd primes. We always choose odd primes for RSA, as otherwise it would be easy to see that one of the primes is 2 and therefore to factor N. So let's assume p and q are both odd primes.

(a) First we show that if p is an odd prime and $\gcd(a,p) = 1$, then $x^2 \equiv a \pmod p$ has either no solution or two solutions. Suppose that it has a solution, call it b. Then it has another solution, namely $-b$. (Note that $b \not\equiv -b \pmod p$, otherwise we would have $2b \equiv 0 \pmod p$ which since p is odd would force $b \equiv 0 \pmod p$. But $b^2 \equiv a \not\equiv 0 \pmod p$, so $b \not\equiv 0 \pmod p$ since $\mathbb{Z}/p\mathbb{Z}$ is a field and doesn't have zero divisors.)

However, $x^2 \equiv a \pmod p$ cannot have more than two solutions. Suppose there were a third solution c (and therefore also a fourth solution $-c$). Choose both b and c such that $0 < b, c < \frac{p}{2}$ (if either b or c does not satisfy this, $-b$ or $-c$ will satisfy this, just switch them out). Then $b^2 \equiv c^2 \equiv a \pmod p$ (after all, b and c are both solutions of $x^2 \equiv a \pmod p$), so

$$\begin{aligned} b^2 - c^2 &\equiv 0 \pmod p \\ (b-c)(b+c) &\equiv 0 \pmod p. \end{aligned}$$

But we have that $0 < b + c < p$, so p does not divide $b + c$ and therefore p must divide $b - c$ or $b \equiv c \pmod p$, and c is not a new solution. We have therefore proved that if p is an odd prime and $\gcd(a,p) = 1$, then $x^2 \equiv a \pmod p$ has either no solution or two solutions.

Now $x^2 \equiv a \pmod N$ is assumed to have a solution. Therefore $x^2 \equiv a \pmod p$ and $x^2 \equiv a \pmod q$ both have solutions too. They each therefore have exactly two solutions, say b and $-b$ are solutions to $x^2 \equiv a \pmod p$ and c and $-c$ are solutions to $x^2 \equiv a \pmod q$. By the Chinese Remainder Theorem, this gives four solutions to $x^2 \equiv a \pmod N$: The solution that lifts $x \equiv b \pmod p$ and $x \equiv c \pmod q$, the one that lifts $x \equiv b \pmod p$ and $x \equiv -c \pmod q$, the one that lifts $x \equiv -b \pmod p$ and $x \equiv c \pmod q$ and finally the one that lifts $x \equiv -b \pmod p$ and $x \equiv -c \pmod q$.

(b) In complete contradiction with our notation above, let the four solutions of $x^2 \equiv a \pmod N$ be $b, -b, c$ and $-c$. Then we have that

$$b^2 \equiv c^2 \equiv a \pmod N,$$

or

$$b^2 - c^2 \equiv (b-c)(b+c) \equiv 0 \pmod N.$$

It suffices to show now that $\gcd(b-c, N) = p$ and $\gcd(b+c, N) = q$ (or vice versa). Note that since $(b-c)(b+c) \equiv 0 \pmod N$, we know that

$$\gcd(b-c, N) > 1 \quad \text{or} \quad \gcd(b+c, N) > 1,$$

as in problem 3 above. The issue here is to prove that N does not divide $b-c$ or $b+c$. If this were the case the gcd computation would just give us back N and not a factor of N.

However, we know that N does not divide $b-c$, because we have assumed that b and c are different modulo N. In the same way, we know that N does not divide $b+c$ because we have assumed that b and $-c$ are different modulo N. Therefore we know that N divides neither $b+c$ nor $b-c$ and so that p must divide one and q the other for their product to be zero modulo N.

Note that this, in retrospect, shows that we did not get lucky in problem 3. Since $xy \not\equiv 14 \pmod N$, we could have known in advance that the gcd would yield a nontrivial factor of N.

3. (a) Write $a \equiv g^A \pmod p$, $b \equiv g^B \pmod p$ and $ab \equiv g^C \pmod p$, where $0 \le A, B, C < p-1$ (this is possible since g is a primitive root of p). We have therefore that $g^C \equiv g^A g^B \equiv g^{A+B} \pmod p$. Since g is a primitive root of p, we have that $g^{p-1} \equiv 1 \pmod p$, but no lower power of g is congruent to 1 modulo p. In particular, this means that $g^C \equiv g^D \pmod p$ if and only if $C \equiv D \pmod{p-1}$ (one direction is because $g^{p-1} \equiv 1 \pmod p$, and the other is because $g^k \not\equiv 1 \pmod p$ for any $0 < k < p-1$). Therefore

$$C \equiv A + B \pmod{p-1}$$

or in other symbols,

$$\log_g(ab) \equiv \log_g(a) + \log_g(b) \pmod{p-1}.$$

(b) Let $p = 7$, then 6 is not a primitive root modulo 7. We also have

$$2 \equiv \log_6 6 + \log_6 6 \not\equiv \log_6(36) \equiv \log_6 1 \equiv 0 \pmod 6.$$

The correct equation is that, if $\operatorname{ord}_p a$ is the multiplicative order of a modulo p (i.e. the smallest positive integer k such that $a^k \equiv 1 \pmod{p}$), then

$$\log_g(ab) \equiv \log_g(a) + \log_g(b) \pmod{\operatorname{ord}_p a}.$$

It just so happens that if g is a primitive root of p, then $\operatorname{ord}_p g = p - 1$, by definition.

(c) i. We have that $g^0 \equiv 1 \pmod{p}$ and $g^1 \equiv g \pmod{p}$, so the result follows by the definition of $\log_g$.

ii. Since $aa^{-1} \equiv 1 \pmod{p}$, taking $\log_g$ on both sides and using parts (a) and (c)i.we get

$$\log_g a + \log_g(a^{-1}) \equiv 0 \pmod{p-1},$$

or $\log_g(a^{-1}) \equiv -\log_g a \pmod{p-1}$.
Now we can prove the general power formula: If $r > 0$, the formula follows by repeated application of part (a). If $r = 0$, the formula follows by part (c)i. And if $r < 0$, the formula follows by writing $a^r \equiv a^{-1} \cdot \ldots \cdot a^{-1} \pmod{p}$, where a^{-1} appears $-r$ times and applying parts (a) and the formula $\log_g(a^{-1}) \equiv -\log_g a \pmod{p-1}$ proved above.