crypto pkc handout: show all work

1. Convert decimal 97 to binary:

   Convert binary `0b 111011` to decimal:

2. • Find the exponent e such that the number printed equals a^e.
   • Shuffle the code so that the number printed equals a^{19}.
   • How many lines of similar code are needed if e has 500 decimal digits? Hint: converting from decimal to binary increases digits by a factor of $\lg(10) \approx 10/3$.

   ```
   x = 1
   x = x*x*a
   x = x*x*a
   x = x*x
   x = x*x
   x = x*x*a
   print x
   ```

3. Alice and Bob use DHM and email to create a secret key. Give an example of their first email exchange.

   Assume they use base 7 and modulus 11, Bob's secret exponent is 3, and Alice's secret exponent is unknown. What number does Bob send Alice?

   Alice sends Bob the number 4. What secret key do Bob and Alice create?

   You are Eve. Find Alice's exponent. Find Alice and Bob's secret key.

4. Let x, y, n be 1234567809, 12345, 9087654321. My laptop can perform 1 64-bit integer mod operation in 1 microsecond. Estimate the number of seconds needed for each of the following.

   find $x^y(\mod n)$

   find t such that $x^t = 2672633475(\mod n)$

5. Alice's RSA key has $n = 77$ and $e = 7$. Bob encrypts secret message $m = 9$ and sends it to Alice. Eve wants to find m.

   What number β does Bob send Alice?

   Can Eve see β?

   How does Alice recover m from β?

   Show that the number that Alice computes is equal to m.

   How does Eve find m?

   For real world RSA, how long would it take Eve to find m?

1.

```
97 1                          check answer
48 0                          0b 1 = dec 1
24 0                          0b 11 = dec 2*(1) + 1 = 3
12 0                          0b 110 = dec 2*(3) = 6
 6 0                          0b 1100 = dec 2*(6) = 12
 3 1                          0b 11000 = dec 2*(12) = 24
 1 1                          0b 110000 = dec 2*(24) = 48
so decimal 72 = 0b 1100001    0b 1100001 = dec 2*(48) + 1 = 97

0b 1 = dec 1                  exercise: check answer
0b 11 = dec 2*(1) + 1 = 3
0b 111 = dec 2*(3) + 1 = 7
0b 1110 = dec 2*(7) = 14
0b 11101 = dec 2*(14) + 1 = 29
0b 111011 = dec 2*(29) + 1 = 59
```

2.

```
x = 1
x = x*x*a   = a                   1
x = x*x*a   = a*a*a = a^3         1
x = x*x     = (a^3)*(a^3) = a^6   0
x = x*x     = a^12                0
x = x*x*a   = a^25                1  check: 0b 11001 = dec 25

dec 19 = 0b 10011, so
x = 1
x = x*x*a   = a
x = x*x     = a^2
x = x*x     = a^4
x = x*x*a   = a^9
x = x*x*a   = a^19
```

e has about $500\times$ (10/3) $\approx$ 1700 bits, so about 1700 similar lines of code

3. Alice: hey. Bob: hey. A: let's create a DHM key. B: ok. A: so, what prime should we use? B: 11? A: ok. how about 7 for the base? B: ok. A: ok, ciao. B: later.

let g, n be the base and modulus. Bob sends $\beta = g^b \pmod{n} = 7^3 \pmod{11} = 2$.

they create $g^{ab} \pmod{n} = (g^a)^b \pmod{n} = \alpha^b \pmod{n} = 4^3 \pmod{n} = 9$.

Eve knows g, n, α, β. Because the numbers here are small, she can solve the discrete log problem: find a so that $g^a = \alpha \pmod{n}$, ie. so that $7^a = 4 \pmod{11}$. Using python `pow` function, she finds that the powers of 7 (mod 11) are respectively 1,7,5,2,3,10,4,6,9,8,1. So she sees that $a = 6$. Now she can compute the secret key $g^{ab} \pmod{n} = (g^b)^a \pmod{n} = 2^6 \pmod{11} = 9$.

4. $n = 9087654321$ has 10 decimal digits, so about (10/3)*10 = 33 bits, which is less than 64. So my laptop can perform 1 multiplication mod n in 1 microsecond.

   y = 12345 has about 12345×(10/3)=4×10^4 bits. The number of modular multiplications to raise x to the power y is at most 3 times this number, so about 1.2×10^5, so this many microseconds, so about .1 seconds.

   best known algorithm is not much better than brute force search. so we need to try all possible exponents. there are 2672633475 of them. each try takes about .12 seconds. so about 3×10^8 seconds, or about 10 years. (on a fast computer, this problem would be tractable. there are machines today capable of petaflops, ie. 10^{15} floating point operations per second).

5. Bob sends Alice $\beta = m^e \pmod{n} = 9^7 \pmod{77} = 37$.

   yes

   Alice computes $\beta^d \pmod{n}$, where $d = e^{-1} \pmod{\phi(n)}$.

   mod n, $\beta^d = (m^e)^d = m^{ed} = m^{k*\phi(n)+1} = (m^{\phi(n)})^k * m^1 = 1^k * m^1 = m$

   Eve can either factor n into primes p, q or find d without factoring n. Here, it is easy to factor $n = 7 * 11$, so $\phi(n) = 6 * 10 = 60$. Now use extended Euclid gcd algorithm with $e = 7$ and $\phi(n) = 60$:

```
60 = 7*8 + 4        4 = 60 - 7*8
 7 = 4*1 + 3        3 = 7 - 4
 4 = 3*1 + 1        1 = 4 - 3
 3 = 1*3 + 0

1 = 4 - 3*1         = 4 - (7 - 4*1)*1
  = 2*4 - 7         = 2*(60 - 7*8) - 7
  = 2*60 - 17*7
```

   so 7 inverse mod 60 is $-17 = 43$, so $d = 43$. So Eve can compute $\beta^d \pmod{n} = 37^43 \pmod{77} = 9$.

6. presumably, depending on the RSA settings, and the computing resources available, many many years