Homework 4

1. **An Example of Extended GCD Algortihm (20 points).** Recall that the extended GCD algorithm takes as input two integers a, b and returns a triple (g, α, β), such that

$$g = \gcd(a, b), \text{ and } g = \alpha \cdot a + \beta \cdot b.$$

Here $+$ and $\cdot$ are integer addition and multiplication operations, respectively.

Find (g, α, β) when $a = 310, b = 2020$.

The following calculations are based on this code snippet:

```
XGCD(A,B):
   if B = 0:
     return (A,1,0)

   R = A%B
   M = (A-R)/B
   (G, a', B') = XGCD(B,R)

   return (G, B', a' - B' * M)
```

Note that this is a recursive algorithm. The functions executed are listed largest to smallest. For simplicity, I calculate XGCD(2020,310) instead. This will be adjusted later.

```
XGCD(2020,310):
   R = 2020%310 = 160
   M = (2020-160)/310 = 6
   (G, a', B') = XGCD(310,160) = (10,-1,2)

   return (G, B', a' - B' * M) = (10,2,-1-(2*6)) = (10,2,-13)


XGCD(310,160):
   R = 310%160= 150
   M = (310-150)/160 = 1
   (G, a', B') = XGCD(160,150) = (10,1,-1)

   return (G, B', a' - B' * M) = (10,-1,1-(-1*1)) = (10,-1,2)
```

```
XGCD(160,150):
   R = 160%150 = 10
   M = (160-10)/150 = 1
   (G, a', B') = XGCD(150,10) = (10,0,1)

   return (G, B', a' - B' * M) = (10,1,0-(1*1)) = (10,1,-1)


XGCD(150,10):
   R = 150%10 = 0
   M = (150-0)/10 = 0
   (G, a', B') = XGCD(10,0) = (10,1,0)

   return (G, B', a' - B' * M) = (10, 0, 1-(0*0)) = (10,0,1)


XGCD(10,0):
   if B = 0:
     return (A,1,0) = (10,1,0)
```

The result is XGCD(2020,310) $=$ (10,2,-13). However, we were supposed to calculate XGCD(310,2020), so we just swap the numbers around.

Therefore, our final answer is that XGCD(310,2020) $=$ (10,-13,2), where $10 = -13 * 310 + 2 * 2020 = -4030 + 4040 = 10$.

2. **(20 points).** Suppose we have a cryptographic protocol P_n that is implemented using αn^2 CPU instructions, where α is some positive constant. We expect the protocol to be broken with $\beta 2^{n/10}$ CPU instructions.

   Suppose, today, everyone in the world uses the primitive P_n using $n = n_0$, a constant value such that even if the entire computing resources of the world were put together for 8 years we cannot compute $\beta 2^{n_0/10}$ CPU instructions.

   Assume Moore's law holds. That is, every two years, the amount of CPU instructions a CPU can run per second doubles.

   (a) (5 points) Assuming Moore's law, how much faster will be the CPUs 8 years into the future as compared to the CPUs now?

   Assuming Moore's law, CPUs will be 16x faster 8 years in the future as compared to CPUs now.

   (b) (5 points) At the end of 8 years, what choice of n_1 will ensure that setting $n = n_1$ will ensure that the protocol P_n for $n = n_1$ cannot be broken for another 8 years?

   To solve this, we need to make sure that we use a value of n_1 such that $16 * \beta 2^{n_0/10} = \beta 2^{n_1/10}$.

   $\beta 2^{n_1/10} = 16 * \beta 2^{n_0/10}$
   $2^{n_1/10} = 16 * 2^{n_0/10}$
   $2^{n_1/10} = 2^{(n_0/10)+4}$
   $2^{n_1/10} = 2^{(n_0+40)/10}$
   $\frac{n_1}{10} = \frac{n_0+40}{10}$
   $n_1 = n_0 + 40$

   Therefore, if we set $n_1 = n_0 + 40$, the protocol P_n should be secure for another 8 years.

(c) (5 points) What will be the run-time of the protocol P_n using $n = n_1$ on the new computers as compared to the run-time of the protocol P_n using $n = n_0$ on today's computers?

We can set $n_1 = n_0 + 40$, and we know that the number of instructions executed (run-time) for $P_n = \alpha n^2$ for some n.

Since we see that the run-time protocol P_n using $n = n_0$ on today's computers is $\alpha * {n_0}^2$, it follows that the protocol P_n using $n = n_1$ on today's computers has run-time $\alpha * {n_1}^2$. Since $n_1 = n_0 + 40$, the new protocol runs, in terms of n_0, in time $\alpha * (n_0 + 40)^2$ on today's computers. However, since the new protocol runs on the new computers, which are 16x faster, we divide the run-time by 16. Therefore, the final run-time for the new protocol on the new computers, in terms of n_0, is $(\alpha(n_0 + 40)^2)/16$.
Taking the ratio between the two gives us the following equation:

$\frac{(\alpha(n_0+40)^2)/16}{\alpha {n_0}^2}$

$= \frac{n_0^2+80n_0+1600}{16*n_0^2}$

(d) (5 points) What will be the run-time of the protocol P_n using $n = n_1$ on today's computers as compared to the run-time of the protocol P_n using $n = n_0$ on today's computers?

This part uses the same logic as part (c). I've already stated the following (on today's computers, and in terms of n_0):

i. The old protocol's run-time is $\alpha * {n_0}^2$.
ii. The new protocol's run-time is $\alpha * (n_0 + 40)^2 = \alpha({n_0}^2 + 80n_0 + 1600)$.

If we take the ratio between the two, we get the following:

$\frac{\alpha*(n_0+40)^2}{\alpha*{n_0}^2}$

$= \frac{{n_0}^2+80n_0+1600}{{n_0}^2}$

(*Remark*: This problem explains why we demand that our cryptographic algorithms run in polynomial time and it is exponentially difficult for the adversaries to break the cryptographic protocols.)

3. **Finding Inverse Using Extended GCD Algorithm (20 points).** In this problem we shall work over the group $(\mathbb{Z}_{503}^*, \times)$. Note that 503 is a prime. The multiplication operation $\times$ is "integer multiplication $\mod 503$."

   Use the Extended GCD algorithm to find the multiplicative inverse of 50 in the group $(\mathbb{Z}_{503}^*, \times)$.

   First off,
   XGCD(X,P) = XGCD(50,503) = (1,-171,17)

   The proof used in class states that:

   $1 = \alpha * X + \beta * P$
   $1 = \alpha * X + \beta * 0 \pmod p$
   $1 = \alpha * X \pmod p$

   So $a \mod p$ **is the multiplicative inverse of** X **in the group.**

   Therefore, $mult.inv(50) = -171 \mod 503 = 332$**.**

4. **Another Application of Extended GCD Algorithm (20 points).** Use the Extended GCD algorithm to find $x \in \{0, 1, 2, \ldots, 1538\}$ that satisfies the following two equations.

$$x = 10 \mod 19$$
$$x = 7 \mod 81$$

Note that 19 is a prime, but 81 is <u>not</u> a prime. However, we have the guarantee that 19 and 81 are relatively prime, that is, $\gcd(81, 19) = 1$. Also note that the number $1538 = 19 \cdot 81 - 1$.

First off, we can calculate that $XGCD(81, 19) = (1, 4, -17) \rightarrow 81 * 4 - 19 * -17 = 1$

We can also see that $81 * 4 = 324 = 1 \mod 19$**. Therefore, 4 is 81's inverse** $\mod 19$**. Also,** $19 * -17 = -323 = 1 \mod 81$**. Therefore, -17 is 19's inverse** $\mod 81$**.**

Using those deductions, we know that:
$10 * (81 * 4) = 10 \mod 19$
and
$7 * (19 * -17) = 7 \mod 81$

Finally, from HW3 5(c), we concluded that $x = a * x_p + b * x_q$ **satisfies** x **for** $x \pmod p = a$ **and** $x \pmod q = b$**. Plugging what we have into that equation, we get the following:**

$x = a * x_p + b * x_q$
$x = 10 * (81 * 4) + 7 * (19 * -17)$
$x = 3240 - 2261$
$x = 979 \mod 1539$
$x = 979$

Therefore, we can conclude that the $x \in \{0, 1, 2, \cdots, 1538\}$ **that satisfies** $x = 10 \mod 19$ **and** $x = 7 \mod 81$ **is** 979**.**

5. **Square Root of an Element (20 points).** Let p be a prime such that $p = 3 \mod 4$. For example, $p \in \{3, 7, 11, 19 \dots\}$.

   We say that x is a square-root of a in the group $(\mathbb{Z}_p^*, \times)$ if $x^2 = a \mod p$. We say that $a \in \mathbb{Z}_p^*$ is a quadratic residue if $a = x^2 \mod p$ for some $x \in \mathbb{Z}_p^*$. Prove that if $a \in \mathbb{Z}_p^*$ is a quadratic residue then $a^{(p+1)/4}$ is a square-root of a.

   (Remark: This statement is only true if we assume that a is a quadratic residue. For example, when $p = 7$, 3 is not a quadratic residue, so $3^{(7+1)/4}$ is not a square root of 3.)

   If $(a^{(p+1)/4})^2 = a \mod p$, we can say that $a^{(p+1)/4}$ is a square root of a.

   We can start by squaring $a^{(p+1)/4}$:

   $(a^{(p+1)/4})^2$
   $a^{(p+1)/2}$

   We also know that a is a quadratic residue, so $a = x^2 \mod p$ for some $x \in Z_p^*$. Substituting that into what we had above,

   $(x^2)^{(p+1)/2} \pmod p$
   $x^{(p+1)} \pmod p$

   Splitting that equation apart, and remembering that $x^p = x \mod p$ (which we proved in previous homeworks),

   $x^{(p+1)} \pmod p$
   $x^p * x \pmod p$
   $x * x \pmod p$
   $x^2 \pmod p$

   $(a^{(p+1)/4})^2 = x^2 \pmod p$

   And since $a = x^2 \mod p$,

   $(a^{(p+1)/4})^2 = a \pmod p$

   Therefore, we can confidently conclude that $a^{(p+1)/4}$ is a square root of a.

Collaborators: Angga, Benjamin