# 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 $(g, \alpha, \beta)$, 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 $(g, \alpha, \beta)$ when $a = 310, b = 2020$.

**Solution.**

$$\begin{aligned} 2020 &= 310 \times 6 + 160 \\ 310 &= 160 \times 1 + 150 \\ 160 &= 150 \times 1 + 10 \\ 150 &= 10 \times 15 + 0 \end{aligned}$$

Therefore, we have:

$$\begin{aligned} 10 &= \mathbf{160} \times 1 + \mathbf{150} \times (-1) \\ &= 160 + (310 - 160 \times 1) \times (-1) \\ &= \mathbf{310} \times (-1) + \mathbf{160} \times 2 \\ &= 310 \times (-1) + (2020 - 310 \times 6) \times 2 \\ &= \mathbf{2020} \times 2 + \mathbf{160} \times (-13) \end{aligned}$$

Answer: $\gcd(2020, 310) = 10$, $\alpha = -13$, and $\beta = 2$.

2. **(20 points).** Suppose we have a cryptographic protocol $P_n$ that is implemented using $\alpha n^2$ CPU instructions, where $\alpha$ 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?
   The CPUs will be $2^{8/2} = 16$ times faster.

   (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?
   Currently, it is given that $\frac{\beta 2^{n_0/10}}{\Lambda} =$ 8-years. We want $\frac{\beta 2^{n_1/10}}{16\Lambda} =$ 8-years. That is

$$16 \cdot 2^{n_0/10} = 2^{n_1/10} \iff n_1 = n_0 + 40$$

(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? Today, suppose, a honest person runs $\Gamma$ instructions per second. The run-time of $P_{n_0}$ on today's computers is

$$\frac{\alpha n_0^2}{\Gamma}$$

8-years from now, honest people will run $16\Gamma$ instructions per second on new computers. The run-time of $P_{n_1}$ on those computers is

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

The ratio of second-run-time to first-run-time is

$$\left(\frac{1}{4} + \frac{10}{n_0}\right)^2$$

Observe: This ratio can be much less than 1. That is, in future, honest people will be able to run the protocol faster!

(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? Suppose the honest person did not upgrade his CPU. So, he is running $\Gamma$ instructions per second in the future as well. Then, the running time of $P_{n_1}$ on this computer is

$$\frac{\alpha n_1^2}{\Gamma} = \frac{\alpha(n_0+40)^2}{\Gamma}$$

Ratio of this time to run-time of $P_{n_0}$ is

$$\frac{(n_0+40)^2}{n_0^2} = \left(1 + \frac{40}{n_0}\right)^2$$

Observe: If $n_0$ is much larger than 40 then the running-time of $P_{n_1}$ on old processors is not much different from running-time of $P_{n_0}$ on old processors!

(*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 $(\mathbb{Z}_{503}^*, \times)$.

   **Solution.**
   First, we have:

$$\begin{aligned} 503 &= 50 \times 10 + 3 \\ 50 &= 3 \times 16 + 2 \\ 3 &= 2 \times 1 + 1 \end{aligned}$$

   Then, we have the following:

$$\begin{aligned} 1 &= \mathbf{3} \times 1 + \mathbf{2} \times (-1) \\ &= \mathbf{3} \times 1 + (50 - 3 \times 16) \times (-1) \\ &= \mathbf{50} \times (-1) + \mathbf{3} \times (17) \\ &= \mathbf{50} \times (-1) + (503 - 50 \times 10) \times (17) \\ &= \mathbf{503} \times (17) + \mathbf{50} \times (-171) \end{aligned}$$

   Now, we have:

$$1 = \mathbf{503} \times (17) + \mathbf{50} \times (-171) \mod 503 = 0 + 50 \times (-171 \mod 2017) = 50 \times 332 \mod 503$$

   So, the inverse of 50 modulo 503 is **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.

$$\begin{aligned} x &= 10 \mod 19 \\ x &= 7 \mod 81 \end{aligned}$$

Note that 19 is a prime, but 81 is not 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 $1538 = 19 \cdot 81 - 1$.

**Solution.**

$$\begin{aligned} 81 &= 19 \times 4 + 5 \\ 19 &= 5 \times 3 + 4 \\ 5 &= 4 \times 1 + 1 \end{aligned}$$

So, we have the following:

$$\begin{aligned} 1 &= \mathbf{5} \times 1 + \mathbf{4} \times (-1) \text{ an integer linear combination of} \\ &= 5 \times 1 + (19 - 5 \times 3) \times (-1) \\ &= \mathbf{5} \times (4) + \mathbf{19} \times (-1) \\ &= (81 - 19 \times 4) \times (4) + 19 \times (-1) \\ &= \mathbf{81} \times 4 + \mathbf{19} \times (-17) \end{aligned}$$

This implies to have the following:

$$\begin{aligned} \mathbf{81} \times 4 &\mod 19 = 1 \\ \mathbf{19} \times (-17) &\mod 81 = 1 \end{aligned}$$

Now, we claim that $\mathbf{81} \times 4 \times 10 + \mathbf{19} \times (-17) \times 7 = 979$ satisfies both equations. The reason is the following:

$$\mathbf{81} \times 4 \times 10 + \mathbf{19} \times (-17) \times 7 \mod 19 = (\mathbf{81} \times 4 \mod 19) \times (10 \mod 19) + 0 = 1 \times 10 = 10$$
$$\mathbf{81} \times 4 \times 10 + \mathbf{19} \times (-17) \times 7 \mod 81 = 0 + (\mathbf{19} \times (-17) \mod 81) \times (7 \mod 81) 1 \times 7 = 7$$

Answer: **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.)

**Solution.**
Since $a \in \mathbb{Z}_p^*$ is a quadratic residue, we have $a = x^2 \mod p$ for some $x \in \mathbb{Z}_p^*$. Then, since $x \in \mathbb{Z}_p^*$, we have $x^{p-1} \mod p = 1$, and so:

$$a^{\frac{(p+1)}{4}} = x^{\frac{p+1}{2}} \mod p$$
$$\implies \left(a^{\frac{(p+1)}{4}}\right)^2 = \left(x^{\frac{p+1}{2}}\right)^2 = x^{p+1} = x^{p-1} \times x^2 = 1 \times a = a$$

This proves that $a^{\frac{p+1}{4}}$ is a square-root of $a$.

**Collaborators :**