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

\[
g = \gcd(a, b), \quad g = \alpha \cdot a + \beta \cdot b.
\]

Here + and \( \cdot \) are integer addition and multiplication operations, respectively. Find \((g, \alpha, \beta)\) when \( a = 2022, b = 125 \). You must show all your work.

**Solution.**
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?

Solution.

(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?

Solution.
(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?

Solution.

(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?

Solution.

(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 \((\mathbb{Z}_{257}^*, \times)\). Note that 257 is a prime. The multiplication operation \(\times\) is “integer multiplication \(\mod 257\).”

Use the Extended GCD algorithm to find the multiplicative inverse of 55 in the group \((\mathbb{Z}_{257}^*, \times)\). You must show all your work.

**Solution.**
4. Another Application of Extended GCD Algorithm (20 points). Use the Extended GCD algorithm to find \( x \in \{0, 1, 2, \ldots, 2235\} \) that satisfies the following two equations.

\[
\begin{align*}
    x &\equiv 2 \pmod{52} \\
    x &\equiv 3 \pmod{43}
\end{align*}
\]

Note that 43 is a prime, but 52 is not a prime. However, we have the guarantee that 52 and 43 are relatively prime, that is, \( \gcd(52, 43) = 1 \). Also note that the number \( 2235 = 52 \cdot 43 - 1 \).

You must show all your work.

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

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.
Collaborators: