MCS 425: Codes and Cryptography (Fall 2020)

Homework 2

Due at 12:00pm CDT, Friday, Oct 2

1. Compute $16^{-1} \bmod 27$ by hand using the extended Euclidean algorithm. Show your work.

2. Let a, b, n, and d be positive integers. Prove that $ad \equiv bd \pmod{nd}$ if and only if $a \equiv b \pmod{n}$.

3. Solve x in $48x \equiv 30 \pmod{81}$ by hand. Show your work.

4. Compute $5^{119} \bmod 36$ by hand. Show your work.

5. (2 points) In this exercise, we will use Euler's theorem and the Chinese remainder theorem to speed up the computation of $(a^x \bmod n)$ when $\gcd(a, n) > 1$.

   a) Compute $10^{130} \bmod 3$ using Euler's theorem. Compute $10^{130} \bmod 16$.

   b) Compute $10^{130} \bmod 48$ from (a) using the Chinese remainder theorem.

6. Implement the extended Euclidean algorithm and fast modular exponentiation using repeated squaring. Use your code to compute $(9007^{-1} \bmod 305612224)$ and $(20019^{254071759} \bmod 305647189)$.

   (Your only need to answer two integers for this question.)