Problem Set 3

Due: March 26

1. Suppose that plaintext message units are digraphs in the ordinary 26-letter alphabet, where (as usual) A-Z corresponds to 0-25. We represent each digraph ij as an integer in the following way:

$$(26 \cdot i) + j.$$

For example, the digraph "IN" is represented as

$$26 \cdot 8 + 13 = 221.$$

You receive the sequence of ciphertext message units,

$$2723, 3532, 6132, 5713, 10008, 4682, 2816, 5762, 2940$$

which were encrypted using a Merkle-Hellman knapsack system. (Each unit is the encryption of a digraph, and hence represents two plaintext letters.) The private key for the system is

$$(M = 2647, W = 1036, \{3, 5, 11, 20, 41, 83, 165, 329, 662, 1321\}).$$

Your job is to decipher the message.

2. Use the repeated squaring method (Algorithm 3.3 in the number theory notes) to calculate 652^{853} mod 4847.

In your proofs for problems 3-7, you may use without proof any result stated in sections 1-3 of the Number Theory notes (except, of course, if the given result itself is what you're being asked to prove. In that case, you may use without proof any result stated in sections 1-3 of the Number Theory notes **up to** *the result I'm asking you to prove.) If you use some number theoretic result not stated in sections 1-3 of the notes, you should prove that result first (using, of course, the results in sections 1-3 of the notes).*

3. Complete the proof of the *gcd* recursion theorem (Theorem 1.6 in the number theory notes). We have already proved that $gcd(a, b) \leq gcd(b, a \bmod b)$. So I am asking you here to prove

$$gcd(b, a \bmod b) \leq gcd(a, b).$$

4. Complete the proof of Theorem 1.13 in the notes: If there are integers x, y such that $ax + by = 1$, then $gcd(a, b) = 1$.

5. Prove the following: If $a|c$, $b|c$, and $gcd(a, b) = 1$, then $ab|c$.

6. Prove that if $m \perp n$, $a \equiv b \pmod{m}$, and $a \equiv b \pmod{n}$, then $a \equiv b \pmod{mn}$

7. Let a be a non-negative integer, b, c, M positive integers. Let $d = gcd(c, M)$. Prove:

$$\text{If there exists a } k \text{ such that } a + kc \equiv b \bmod M, \text{ then } d|(b - a).$$

[**Note:** This problem arose in a project on formal software analysis I worked on a few years ago. Consider a software **for** loop of the form

```
for ($i = a$; $i \neq b$; $i = i + c$) {
    Body of Loop;
},
```

where a is a non-negative integer and b and c positive integers. Consider the question of whether this loop will ever terminate. This question isn't as trivial as you might think, since computer arithmetic is performed modulo M for some large integer M (e.g., $M = 2^{32}$). Thus i takes the values $a, a + c, a + 2c, \ldots$, where each term in the sequence is reduced modulo M. The value of i might miss b the first time through the loop as it increases towards M, but then hit b later after it cycles around a few times. As an example, consider the case where $a = 3, b = 7, c = 6, M = 16$. Then i takes on the successive values $3, 9, 15, 5, 11, 1, 7$.

To say that the loop terminates is to say that there is some integer k such that, after k iterations, the value of i modulo M is equal to b:

$$a + kc \equiv b \bmod M.$$

So what I'm asking you to prove boils down to the following: if the loop terminates, then $d|(b - a)$. In fact the implication goes both ways: $d|(b - a)$ implies that the loop terminates. This direction is harder to prove.]

8. Stinson, problem 6.13, p.247. Only decipher the first three ciphertext message units in Table 6.1: 12423, 11524, 7243

9. Stinson, problem 6.16, p.250. (When Stinson says, "encrypt each residue modulo 26 as a separate plaintext character", he just means that each letter A-Z is encrypted separately.)