ECE Advanced Information and Network Security
Homework 1 and 2 Solutions
Spring 2007

Problems from the Book:

Chapter 2

2. Changing the plaintext to numbers yields 7, 14, 22, 0, 17, 4, 24, 14, 20. Applying $5x + 7$ to each yields $5 \cdot 7 + 7 = 42 \equiv 16 \pmod{26}$, $5 \cdot 14 + 7 = 77 \equiv 25$, etc. Changing back to letters yields *QZNHOBXZD* as the ciphertext.

4. Let the decryption function be $x = ay + b$. The first letters tell us that $7 \equiv a \cdot 2 + b \pmod{26}$. The second letters tell us that $0 \equiv a \cdot 17 + b$.Subtracting yields $7 \equiv a \cdot (-15) \equiv 11a$. Since $11^{-1} \equiv 19 \pmod{26}$, we have $a \equiv 19 \cdot 7 \equiv 3 \pmod{26}$. The first congruence now tells us that $7 \equiv 3 \cdot 2 + b$, so $b = 1$. The decryption function is therefore $x \equiv 3y + 1$. Applying this to *CRWWZ* yields *happy* for the plaintext.

5. Let $mx + n$ be one affine function and $ax + b$ be another. Applying the first then the second yields the function $a(mx+n)+b = (am)x+(an+b)$, which is an affine function. Therefore, successively encrypting with two affine functions is the same as encrypting with a single affine function. There is therefore no advantage of doing double encryption in this case. (Technical point: Since $\gcd(a, 26) = 1$ and $\gcd(m, 26) = 1$, it follows that $\gcd(am, 26) = 1$, so the affine function we obtained is still of the required form.)

15. The number of seconds in 120 years is

$$60 \times 60 \times 24 \times 365 \times 120 \approx 3.8 \times 10^9.$$

Therefore you need to count $10^{100}/(3.8 \times 10^9) \approx 2.6 \times 10^90$ numbers per second!

Chapter 2 Computer Problems

2. Use 'fr=frequency(lcll);' to get a frequency count. Observe that the most common common letter is l, which is 7 places after e. Try shifting back by 7 using 'shift(lcll,-7)' to get the answer 'ans = eveexpectseggsforbreakfast'.

6. a) The message can be found in the file ciphertexts.m under the variable gaat. The following code performs the conversion to 0, 1, 2, 3, and performs the shifting.

```
ind0=find(gaat==A);
ind1=find(gaat==C);
ind2=find(gaat==G);
ind3=find(gaat==T);
vec=gaat; vec(ind0)=0; vec(ind1)=1; vec(ind2)=2; vec(ind3)=3;
vec=mod(vec+1,4);
ind0=find(vec==0);
ind1=find(vec==1);
ind2=find(vec==2);
ind3=find(vec==3);
output=vec;
output(ind0)=A;
output(ind1)=C;
output(ind2)=G;
output(ind3)=T;
output=char(output);
```

The answer is TCCAAGTGTTGGTGCCAACCGGGAGCGACCCTTTCAGAGACTCCGA.

b) The following code assumes that the affine cipher is of the form y = ax+b (mod 4). The parameters a and b should be entered in. The restrictions are that a is relatively prime to 4 which means that a is either 1 or 3.

```
ind0=find(gaat==A); ind1=find(gaat==C); ind2=find(gaat==G); ind3=find(gaat==T);
vec=gaat;
vec(ind0)=0; vec(ind1)=1; vec(ind2)=2; vec(ind3)=3;
```

vec=mod(a*vec+b,4);
ind0=find(vec==0);
ind1=find(vec==1);
ind2=find(vec==2);
ind3=find(vec==3);
output=vec;
output(ind0)=A;
output(ind1)=C;
output(ind2)=G;
output(ind3)=T;
output=char(output);

Chapter 3

1. (a) Apply the Euclidean algorithm to 17 and 101:

$$101 = 5 \cdot 17 + 16$$

$$17 = 1 \cdot 16 + 1.$$

Working back yields $1 = 17 - 16 = 17 - (101 - 5 \cdot 17) = (-1) \cdot 101 + 6 \cdot 17$.

(b) Since $-101 + 6 \cdot 17 = 1$, we have $6 \cdot 17 \equiv 1 \pmod{101}$. Therefore $17^{-1} \equiv 6 \pmod{101}$.

Chapter 4

1. (a) Switch left and right halves and use the same procedure as encryption. The switch the left and right of the final output. Verification is the same as that on pages 99-100.

(b, c) 1st round: $M_0M_1 \to M_1[M_0 \oplus K \oplus M_1]$

2nd round: $[M_0 \oplus K \oplus M_1][M_1 \oplus M_0 \oplus K \oplus M_1 \oplus K] = [M_0 \oplus K \oplus M_1][M_0]$

3rd round: $[M_0][M_0 \oplus K \oplus M_1 \oplus K \oplus M_0] = [M_0][M_1]$, which is the plaintext.

Therefore 3 rounds is very insecure! After 2 rounds, the ciphertext alone lets you determine M_0 and therefore $M_1 \oplus K$, but not M_1 or K individually. If you also know the plaintext, you know M_1 are therefore can deduce K.

3. CBC: We have $D_K(C_j) \oplus C_{j-1} = D_K(E_K(P_j \oplus C_{j-1})) \oplus C_{j-1} = P_j \oplus C_{j-1} \oplus C_{j-1} = P_j$.

CFB: $C_j \oplus L_8(E_K(X_j)) = (P_j \oplus L_8(E_K(X_j))) \oplus L_8(E_K(X_j)) = P_j$.

4. Let I denote the string of all 1's. Note that the expansion $E(\overline{R_{i-1}}) = \overline{E(R_{i-1})} = E(R_{i-1}) \oplus I$. Therefore $E(\overline{R_{i-1}}) \oplus \overline{K_i} = E(R_{i-1}) \oplus I \oplus K_i \oplus I = E(R_{i-1}) \oplus K_i$, so the input to the S-boxes doesn't change. Therefore the output doesn't change. But $\overline{L_{i-1}} = L_{i-1} \oplus I$, so the resulting right side is $\overline{L_{i-1}} \oplus f(R_{i-1}, K_i) = R_i \oplus I = \overline{R_i}$. Also, clearly the new left side is the complementary string. So each round of DES gives the complementary string, so this is true for the final result.

11. (not assigned, but mentioned as a challenge problem) Let K be the key we wish to find. Use the hint. Then $C_1 = E_K(M_1)$ and $C_2 = E_K(\overline{M_1})$. Now, suppose we start a brute force attack by encrypting M_1 with different keys. If, when we use K_j we get $E_{K_j}(M_1) = C_1$ then we are done and the key we desire is $K = K_j$. However, when we use K_j we can eliminate another key. Here is how. If $E_{K_j}(M_1) = \overline{C_2}$ then we know (by complementation property) that $E_{\overline{K_j}}(\overline{M_1}) = C_2$. Hence, if this happens, we know the key is $\overline{K_j}$ since $\overline{K_j}$ would decrypt C_2 to get $\overline{M_1}$. We are effectively testing two keys for the price of one! Hence, the key space is cut in half and we only have to search an average of 2^{54}.

Chapter 6

1. We have $\phi(n) = (p-1)(q-1) = 100 * 112 = 11200$. A quick calculation shows that $3 \equiv 7467^{-1} \pmod{11200}$. We have $5859^3 \equiv 1415 \pmod{11413}$, so the plaintext was $1415 = no$.

2. (a) Here $\phi(n) = 4 \cdot 10 = 40$. We are looking for a number d such that $ed = 1 \pmod{40}$. Thus, we want to solve for d in $3d = 1 \pmod{40}$. Observe that $d = 27$ gives $3 \cdot 27 = 81 = 1 \pmod{40}$. Hence $d = 27$.

(b) Here, you use Euler's Theorem. d is such that $3d = 1 + k\phi(n)$ for some k. Then, $c^d = m^{3d} = m^{1+k\phi(n)} = m \pmod{n}$ by Euler's Theorem.

8. We have $c_2 \equiv c_1^{e_2} \equiv m^{c_1 c_2} \pmod{n}$. Therefore, this double encryption is the same as single encryption with encryption exponent e_1e_2. So the security is at the same level as single encryption.