ECE Information and Network Security
Exam 1 Solutions
Spring 2005

Instructions: The exam consists of three sections, each section containing several problems. All problems must be solved for maximum points. **Make sure to clearly state which section, and what problem you are solving at the top of each page.**

## 1 Short Answer

1. **(10pts)** Why is encrypting twice with an affine cipher no more effective than encrypting once with an affine cipher? Why is encrypting twice with DES considered a bad strategy for strengthening DES?

   **Answer:** Consider two affine ciphers with keys \((a_1, b_1)\) and \((a_2, b_2)\). Encryption with the first of these is \(y = a_1x + b_1 \pmod{26}\), and similarly for the second. Composition of the two ciphers (encrypting with the first, and then the second) produces \(y = a_2(a_1x + b_1) + b_2 \pmod{26} = a_2a_1x + a_2b_1 + b_2 \pmod{26}\). This is precisely just another affine cipher with key \((a_2a_1, a_2b_1 + b_2)\). So, two affines combined produces another affine, and the key space is not enlarged by double encryption using the affine cipher.

   Although DES is not a group, and double encryption does increase the key space, it is still considered weak because of the meet-in-the-middle attack, which has an effective strength of \(O(2^{56})\).

2. **(10pts)** What is Fermat’s Little Theorem? What is Euler’s Theorem? How do the two theorems differ from each other?

   **Fermat’s Little Theorem:** If \(p\) is a prime and \(a\) is from \(\{1, 2, \ldots, p - 1\}\) then \(a^{p-1} = 1 \pmod{p}\).

   **Euler’s Theorem:** If \(n\) is a composite, and \(a\) is chosen such that \(gcd(a, n) = 1\), then \(a^{\phi(n)} = 1 \pmod{n}\).

   The theorems differ in that Euler’s is the generalization of Fermat’s Little Theorem.

3. **(10pts)** Describe the strategy and the components involved in the EFF Cracker, which was used in 1998 to break DES.

   The EFF Cracker was a single-architecture, parallel strategy. A single computer was connected to 2 chassis. These two chassis were connected to 12 boards, and these 12 boards each had 64 chips on them. Each chip consisted of 24 search units that each would search its own region of the key space. In total, there were about 1500 chips.

   The key to the efficiency of the EFF Cracker was the fact that it used 2 pieces of ciphertexts. Each search unit would try a key on the first ciphertext block, if that decrypted as interesting, it would try the key on the second ciphertext block. If both were interesting, then the search unit would return the key to the main PC software for further checking. A block of ciphertext was decrypted as interesting if it produced all 8 bytes corresponding to letters, numbers, punctuation, etc. For an incorrect key, this happens with probability \(1/65536\). Two ciphertext blocks would both decrypt as interesting with probability \(1/2^{192}\), thereby leaving an effective key space of \(2^{84}\) to be searched by the software.

## 2 Quick Calculations:

1. **(10pts)** Is \(2^{333} + 1\) prime or composite? If composite, find its factors. If prime, explain why.

   **Solution:** This is not prime. It is of the form \(x^3 + y^3 = (x + y)(x^2 - xy + y^2)\). Here, \(x = 2^{111}\) and \(y = 1\).

2. **(10pts)** Suppose your RSA modulus is \(n = 55\), and your encryption exponent is \(e = 3\).

   - **(5pts)** Find the decryption modulus \(d\).
   - **(5pts)** Assume that \(gcd(m, 55) = 1\). Show that if \(c = m^3 \pmod{55}\) is the ciphertext, then the plaintext is \(m = c^d \pmod{n}\). (Do not quote that RSA decryption works, show why it works explicitly using \(d\) and referring to the appropriate theorem).

   **Solution:** (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.
3. **(10pts)** Backwards Bob does not like Feistel Ciphers and decides to create his own style of ciphers. In his scheme, he takes a $2n$ bit message $M$ and divides it into two blocks of $n$ bits, $M = [L_0, R_0]$. During each round $j$ of his algorithm, he will update $L_j$ and $R_j$ via

$$L_j = R_{j-1} \oplus f(L_{i-1}, K_i), \quad R_j = L_{j-1},$$

where $K_j$ is a subkey for the $j$-th round. He will step through $T$ rounds of the algorithm, and declare $C = [L_T, R_T]$ the final ciphertext. Draw a diagram for one round of Bob’s algorithm. Explain how he can decrypt $C$ and show that it works. Is Bob’s scheme all that different from a Feistel-based cipher?

Solution: Observe that this is just the mirror of the usual Feistel Cipher. So, reflect the picture that we are familiar with (so the $f$ box is on the left side). To undo a round (from $[L_j, R_j] \rightarrow [L_{j-1}, R_{j-1}]$, we simply perform:

$$L_{j-1} = R_j,$$

$$R_{j-1} = L_j \oplus f(R_j, K_j) = L_j \oplus f(L_{j-1}, K_j).$$

To decrypt $T$ rounds, we just perform this $T$ times. The scheme is not that different from a Feistel. In fact, its just the same as if we used the normal Feistel, but applied an initial swap of the left and right halves.

## 3 Challenge Questions

1. **(15pts)** Naive Nelson uses RSA to receive a single ciphertext $c$ corresponding to the message $m$. His public modulus is $n$ and his public exponent is $e$. He feels guilty that the system was only used once, so he agrees to decrypt any ciphertext that someone sends him, as long as it is not $c$, and return the answer to that person. Evil Eve sends him the ciphertext $2^e c \pmod{n}$. Explain how Eve can use this to find $m$.

Solution: Nelson decrypts $2^e c$ to get $2^{ed} c^e = 2^d = 2m \pmod{n}$, and therefore sends $2m$ to Eve. Eve divides by $2 \pmod{n}$ (i.e. by applying the inverse of $2 \pmod{n}$) to obtain $m$.

2. Suppose $E^1$ and $E^2$ are two encryption methods (different algorithms). Let $K_1$ and $K_2$ be keys and consider the double encryption

$$E_{K_1, K_2}(m) = E_{K_1}^1(E_{K_2}^2(m))$$

- **(10pts)** Show how to perform a meet-in-the-middle attack on this double encryption.

Solution: This problem was testing to see if you realized that in the meet-in-the-middle attack, it did not matter whether we were using the same algorithm are not.

To perform the meet in the middle attack, you need a plaintext $m$ and ciphertext $c$ pair (its a known plaintext attack). So, make two lists. The left list consists of encryptions using the second encryption $E^2$ with different choices for $K_2$. Similarly, the right side contains decryptions using different keys for the first encryption algorithm. Thus, the lists look like:

| $E_1^2(m)$ | $y_1$ | $z_1 = D_1^1(c)$ |
| $E_2^2(m)$ | $y_2$ | $z_2 = D_2^1(c)$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $E_{788}^2(m)$ | $y_{788}$ | $z_{788} = D_{788}^1(c)$ |
| $\vdots$ | $\vdots$ | $\vdots$ |

Note: The two lists need not be the same size, as the different algorithms might have different key lengths, and hence different amount of keys (see part b). Now, look for matches between $y_j$ and $z_l$. A match using $K_2$ for $E^2$ and $K_1$ for $D^1$ indicates

$$E_{K_2}^2(m) = y = D_{K_1}^1(c)$$

and hence

$$E_{K_1}^1 \left( E_{K_2}^2(m) \right) = c.$$  

- **(10pts)** An affine encryption is $x \rightarrow \alpha x + \beta \pmod{26}$ can be considered a double encryption, where one encryption is multiplying by $\alpha$ and the other is a shift by $\beta$. Show that the meet-in-the-middle attack from above takes at most 38 steps (not including comparison operations). Note that this is much faster than a brute force attack.

Solution: Observe that there are 26 possibilities for $\beta$ and 12 possibilities for $\alpha$. Let $E_2^{\beta}(x) = \alpha x \pmod{26}$ and let $E_3^{\beta}(x) = x + \beta \pmod{26}$. The composition of these two gives the affine cipher. The total computation needed involves producing 26 encryptions for $E^2$ and 12 decryptions for $E^1$. The total is 38.