Homework 2, CSE P 564 -- Computer Security

Out: Nov 1, 2012
Due: Nov 20, 2012 (@11:45pm)

Please submit to Catalyst and include your name and UWNetID on each page of your submission.

(Questions adopted from *Cryptography Engineering*, but you should *not* need the book -- the course textbook and the slides should be sufficient.)

1. Key Strength and Brute Force Attacks (Adopted from *Cryptography Engineering*, 3.8).
Suppose you have a processor that can perform a single DES encryption or decryption operation in 2^{-26} seconds. You also have a known plaintext-ciphertext pair encrypted under an unknown key. How many hours would it take, on average, to find that DES key, using an exhaustive approach:

- With a single processor?
- With a collection of 2^{16} processors?

2. Misusing Stream Ciphers, Part 1 (Adopted from *Cryptography Engineering*, 4.3).
Suppose you, as an attacker, observe the following 32-byte ciphertext C1 (in hex)

46 64 DC 06 97 BB FE 69 33 07 15 07 9B A6 C2 3D
2B 84 DE 4F 90 8D 7D 34 AA CE 96 8B 64 F3 DF 75

and the following 32-byte ciphertext C2 (also in hex)

51 7E CC 05 C3 BD EA 3B 33 57 0E 1B D8 97 D5 30
7B D0 91 6B 8D 82 6B 35 B7 8B BB 8D 74 E2 C7 3B.

Suppose you know these ciphertexts were generated using CTR mode with the same key. Also, they had the same initial counter value (so that the counter inputs to the block cipher when generating C1 is the same as the counter inputs to the block cipher when generating C2). You also know that the plaintext P1 corresponding to C1 is

43 72 79 70 74 6F 67 72 61 70 68 79 20 43 72 79
70 74 6F 67 72 61 70 68 79 20 43 72 79 70 74 6F.

What information, if any, can you infer about the plaintext P2 corresponding to C2? If you can compute P2, then do so; if you can't, state why you can't.

3. CBC Collisions (Adopted from *Cryptography Engineering*, 4.6).
Let P1,P2 be a message that is two blocks long, and let Q1 be a message that is one block long. Let C0,C1,C2 be the encryption of the first plaintext using CBC mode with a random IV and a random key, and let D0,D1 be the encryption of Q1 using CBC mode with a different, random IV and but same key. Recall that using a random IV means that the blocks C0 and D0 are random. Suppose an attacker knows the first message P1,P2 and has intercepted both ciphertexts. Further suppose that, by random chance, D1=C2. Show that the attacker can

compute Q1.

4. CBC-MAC I (Adopted from *Cryptography Engineering*, 6.2).
Suppose c is one block long, a and b are strings that are a multiple of the block length of some block cipher, and $M_K(a \,||\, c) = M_K (b \,||\, c)$, where || denotes string concatenation. Here, M_K is CBC-MAC with a random key K. Explain why the claim that $M_K (a \,||\, d) = M_K (b \,||\, d)$ for any message d is true. You may find it helpful to use one or more figures. Here || denotes string concatenation. Also, the equation is true regardless of the value of K; you do not know the key K.

5. Diffie-Hellman (Adopted from *Cryptography Engineering*, problem 11.4).
Consider the Diffie-Hellman protocol shown in the slide deck.

What problems, if any, could arise if Alice uses the same x and g^x for all her communications with Bob, and Bob uses the same y and g^y for *all* his communications with Alice? Concretely, we suggest writing down what happens the first time Alice and Bob communicates (what the resulting keys are, how those keys are used in some symmetric encryption scheme, and so on). Then do the same for the second time Alice and Bob communicate, and the third and so on. Do you see any problems arising during or after Alice and Bob communicate for the second time.

6. RSA Key Strength (*Cryptography Engineering*, problem 12.7).
Does a 256-bit RSA key (a key with a 256-bit modulus, i.e., n) provide strength similar to that of a 256-bit AES key?

7. RSA Implementation (*Cryptography Engineering*, problem 12.8).
Consider the RSA primitive. Let p = 71, q = 89, and e = 3.

(a) What is n?
(b) What is phi(n)?
(c) The private exponent d is one of these values: 1103, 4107, 5917. Which is it, and how do you know?
(d) Compute the signature on m_1 = 5416, m_2 = 2397, and $m_3 = m_1 m_2 \pmod n$ using the basic RSA operation. Show that the third signature is equivalent to the product of the first two signatures. Please show your work. If you use MATLAB, Wolfram|Alpha, Python, or something similar, please show each command you execute and the resulting response.