Midterm Exam

CS381-Cryptography

November 2, 2012

Useful Items

$||$ denotes concatenation

$\oplus$ denotes exclusive-or

$$2^{10} \approx 10^3 = 1000,$$

and likewise for any (smallish) positive integer k,

$$2^{10 \times k} \approx 10^{3 \times k},$$

so, for instance, 2^{30} is about 10^9, or one billion. (In fact, a 'gigabyte' is not one billion bytes, but 2^{30} bytes.)

1 Exhaustive search attack against symmetric block cipher

Suppose we have a block cipher with a 64-bit key and an 64-bit block size. As usual, we assume that the encryption algorithm E is known to the eavesdropper Eve, but that Eve does not know the key that was used to encrypt the messages she intercepts. We saw in class that it is usually sufficient for Eve to have two ciphertext blocks C_1, C_2 for which she knows the corresponding plaintext blocks P_1, P_2 in order to recover the key through a brute-force attack. (The reason, in a nutshell, is this: While it is quite likely that two different keys for an ideal block cipher encrypt P_1 to C_1, it is extremely unlikely that two different keys encrypt both P_1 to C_1 and P_2 to C_2.)

(a) What kind of attack (apart from 'brute force') is this? (i.e., is it ciphertext-only? chosen plaintext? Something else?)

Solution. Known-plaintext.

(b) How many encryptions of blocks will be performed carrying out this attack? (The number will vary—you might get lucky and discover the key very quickly—but what should you expect roughly?)

Solution. Anything in the neighborhood of 2^{64} is acceptable here. At the very worst, you will have to encrypt P_1under all 2^{64} keys, and whenever you find a match to C_1, check the encryption of P_2 under the same key. This cannot require more than 2^{65} encryptions, but it will in all likelihood find a match in 2^{63} encryptions on average.

(c) Is this attack feasible? That is, is this attack (i) possible to carry out over the course of a day or so on a laptop computer? (ii) within the realm of possibility if one harnesses massive computational resources, including distributed processing and dedicated hardware? (iii) completely outside the realm of possibility?

Solution. The benchmark here is the brute-force attack on DES, with 56-bit keys, that succeeded through dedicated hardware. 2^{64} is 256 times larger, so that is still in the ballpark. The best answer is (ii).

(d) Suppose the block cipher is constructed this way: Both the plaintext block P and the key K are split into two 32-bit subblocks, and each half of the plaintext is encrypted separately with the corresponding half of the key, using a 32-bit block cipher E'. That is,

$$P = P'||P''$$

$$K = K'||K''$$

$$E(K, P) = E'(K', P')||E'(K'', P'').$$

(See Figure 1.) Does this structure change the feasibility of a brute-force attack? Explain.

Solution. Since you are encrypting the two blocks separately, you only need about 2^{32} encryptions to get a match on the first half of the plaintext, and 2^{32} for the second half. So now we are talking in the neighborhood of 2^{33} encryptions, which is possible to carry out in a day on a laptop computer.

2 Block cipher modes of operation

Parts (a) and (b) refer to a 4-bit block cipher with a fixed key K. The encryption function E_K is given by the table on the last page. In part (c), the key is unknown. The problems refer to CBC and CTR mode. For your convenience, block diagrams for these modes of operation are included with the exam. Show all your calculations carefully.

(a) A 2-block plaintext is encrypted using CBC mode with the accompanying table. The three blocks 0101 1110 0010 are received. The first block is the initialization vector. Find the **second plaintext block**.

(b) A 2-block plaintext is encrypted using CTR mode with the accompanying table. The three blocks 0101 1110 0010 are received (the same as in part (a)). The first block is the initial setting CTR of the counter, and the subsequent values $CTR+1, CTR+2$ are used to produce the pad. Find the **second plaintext block**.

(c) A 2-block plaintext is encrypted using CTR mode. (The accompanying table is **not** relevant here.) The three blocks 0101 1110 0010 are received. You do not know the key, but later you learn that the second plaintext block of the original message was 1111. You then intercept a new encrypted message 0110 0001 0001. What is the **first plaintext block**?

(d) The attack in (c) says something important about the proper use of CTR mode, regardless of block size. What is it?

3 Number Theory and RSA

Alice has the following brilliant idea: 'Why should I bother generating *two* primes to create my RSA keys? I can just take a single very large prime p, choose an encryption exponent e relatively prime to $p-1$ and then pick a decryption exponent $d = e^{-1} \bmod p-1$. Everything will work as before.'

(a) Illustrate the key generation phase of Alice's scheme by finding d when $e = 3$ and $p = 17$. (The answer is 11, but I want you to illustrate all the calculations.)

(b) Illustrate decryption in Alice's scheme by decrypting the ciphertext 10, using the same parameters as in (a). (The answer is 3, but I want you to illustrate all the calculations .)

Is it really true that everything works as before? There are two parts to this question. Give brief but precise explanations for your answers:

(c) Does decryption always undo encryption?

(d) Is the new method secure?

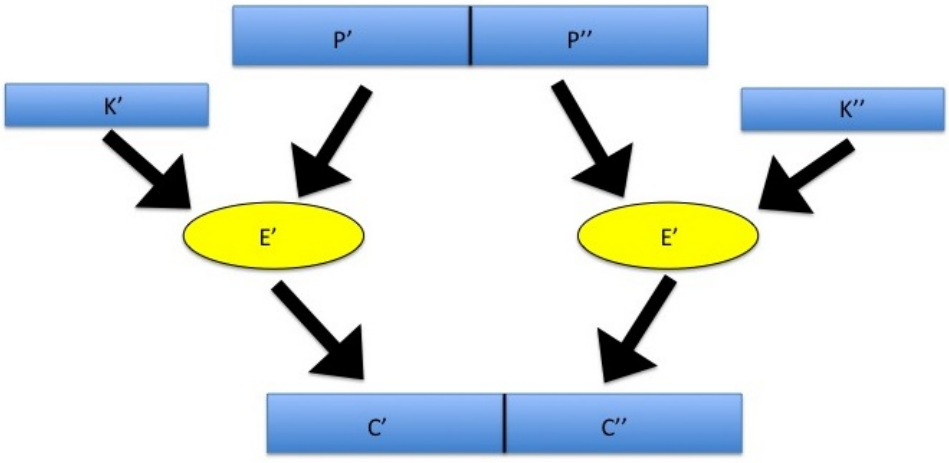

Figure 1: *The structure of the block cipher in Problem 1(d). The two halves of the plaintext block are encrypted separately, each with half the key.*

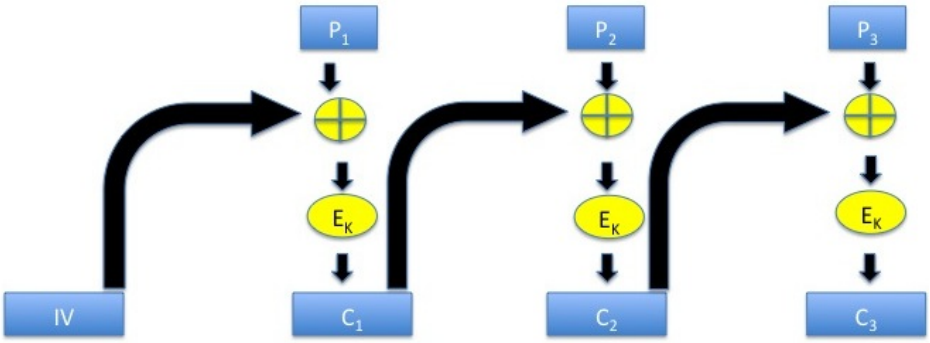

Figure 2: *CBC Encryption.*

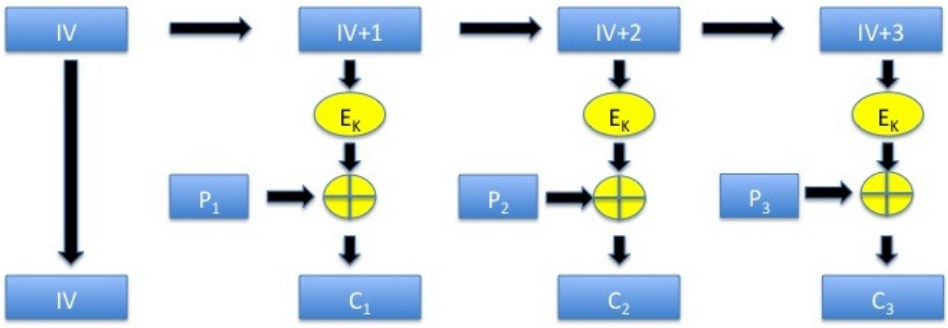

Figure 3: *CTR Encryption.*

P	**E(K,P)**
0000	0011
0001	0001
0010	0111
0011	1100
0100	1001
0101	1111
0110	0000
0111	1010
1000	1101
1001	0010
1010	0101
1011	0110
1100	0100
1101	1000
1110	1011
1111	1110

Table 1: *Block encryption function for the block cipher in 2(a,b), with respect to some fixed key K.*