Assignment #3

Due: January 27, 2011

1) Suppose that you have an n-bit message that is to be encrypted in CBC mode using a block cipher with a m-bit block size. Suppose further that the message length $n > m$ is not a multiple of m, so the message is padded by adding k zeros before encryption ($0 < k < m$). The resulting ciphertext will therefore be k bits longer than the original message.

   In order to be efficient the final k bits are truncated from the penultimate (second to last) block of ciphertext so the resulting ciphertext is now the same length as the original message. Show how the full plaintext can be recovered from the reduced ciphertext – despite the truncated bits.

2) Suppose that a long message has been encrypted in CBC mode and that you now want to decrypt a continuous k-block segment from within the ciphertext. What is the minimum number of blocks of ciphertext that need to be decrypted? If the k-block segment you want to decrypt starts at block i, which blocks do you need to decrypt and how will you decrypt them?

3) Let H be a Merkle-Damgård hash function built out of compression function F. Suppose that you have a black box which can find pre-images of F; that is, the black box takes inputs IV and y and outputs an x such that $F(IV, x) = y$. (You may assume that if you give the black box the same value y multiple times, each time you call it you'll get back a different value x satisfying $F(IV, x) = y$.) Show how by using the black box at most $2k$ times you can find a set of 2^k messages that all have the same hash value when input into the full hash function H.

4) Suppose that you have two hash functions H and H' – each of which produces an n-bit output. You're a little concerned that one or both might be weak, so a co-worker suggests to you that you can defeat any possible attacks by forming a new super-concatenated hash function G with a $2n$-bit output as $G(x) = H(x) \parallel H'(x)$. You and your co-worker are both familiar with the birthday paradox which asserts that if you pick messages at random, you will have to pick 2^n messages before some pair of messages x and y will yield the same $2n$-bit output when fed into G.

   Suppose that your fears are later confirmed and hash function H is badly broken in the sense of the previous problem and an attacker can produce many inputs that all have the same output. How many values would an attacker have to produce in order to find two colliding inputs to the concatenated hash G?

Is your co-worker’s suggestion for $G(x)$ a good idea?

5) Suppose that Alice has a shared symmetric integrity key k with her bank and sends a message to Bob, $m =$ “please pay the bearer \$1”. The message is not encrypted, but its integrity is ensured by using a Merkle-Damgård hash function H with the message integrity code $H(k, m)$. Suppose further that m is an exact multiple of $H's$ block size (so you don’t need to do any padding). Show how Bob, with no knowledge of k, can produce for Alice’s bank a message that is properly integrity-checked with k that instructs that the bearer be paid \$1,000,000.