COMP547B Homework set #4

Due Monday April 2nd, 2018, 23:59:59

Exercises (from Katz and Lindell's book)

5.5 What is the output of an r-round Feistel network when the input is (L_0, R_0) in each of the following two cases:

[5%] (a) Each round function outputs all 0s, regardless of the input.

[5%] (b) Each round function is the identity function.

[10%] 5.6 Show that DES has the property that $DES_k(x) = \overline{DES_{\overline{k}}(\overline{x})}$ for every key k and input x (where $\overline{z}$ denotes the bitwise complement of z). This is called the *complementarity property* of DES. (The description of DES given in this chapter is sufficient for this exercise.)

[10%] 5.8 In the actual construction of DES, the two halves of the output of the final round of the Feistel network are swapped. That is, if the output of the final round is (L_{16}, R_{16}) then the output of the cipher is in fact (R_{16}, L_{16}). Show that the only difference between the computation of DES_k and DES_k^{-1} (given the swapping of halves) is the order of sub-keys.

5.12 This question illustrates an attack on two-key triple encryption. Let F be a block cipher with n-bit block length and key length, and set $F'_{k_1,k_2}(x) = F_{k_1}(F_{k_2}^{-1}(F_{k_1}(x)))$.

[5%] (a) Assume that given a pair (m_1, m_2) it is possible to find in *constant* time all keys k_2 such that $m_2 = F_{k_2}^{-1}(m_1)$. Show how to recover the entire key for F' (with high probability) in time roughly 2^n using three known input/output pairs.

[5%] (b) In general, it will *not* be possible to find k_2 as above in constant time. However, show that by using a pre-processing step taking 2^n time it is possible, given m_2, to find in (essentially) constant time all keys k_2 such that $m_2 = F_{k_2}^{-1}(0^n)$.

[5%] (c) Assume k_1 is known and that the pre-processing step above has already been run. Show how to use a single pair (x, y) for a *chosen* input value x to determine k_2 in constant time.

[5%] (d) Put the above components together to devise an attack that recovers the entire key by running in roughly 2^n time and requesting the encryption of roughly 2^n chosen inputs.

More on back…

5.9 (This exercise assumes the results of the previous exercise.) (**i.e. 5.8**)

[5%] (a) Show that for $k = 0^{56}$ it holds that $DES_k(DES_k(x)) = x$. Why does the use of such a key pose a security threat?

[5%] (b) Find three other DES keys with the same property. These keys are known as *weak keys* for DES.

[5%] (c) Does the existence of these 4 weak keys represent a serious vulnerability in DES? Explain your answer.

5.10 Describe attacks on the following modifications to DES:

[5%] (a) Each round sub-key is 32 bits long, and the mangler function simply XORs the round sub-key with the input to the round (i.e., $\hat{f}(k, R) = k_i \oplus R$). For this example, the key schedule is unimportant and you can treat the k_i as independent keys.

[5%] (b) Instead of using different sub-keys in every round, the same 48-bit sub-key is used in every round. Show how to distinguish the cipher from a random permutation without a 2^{48}-time brute-force search.

Hint: Exercises 5.8 and 5.9 may help...

4.7 Let F be a pseudorandom function. Show that each of the following MACs is insecure, even if used to authenticate fixed-length messages. (In each case Gen outputs a uniform $k \in \{0,1\}^n$. Let $\langle i \rangle$ denote an $n/2$-bit encoding of the integer i.)

[5%] (a) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^n$, compute $t := F_k(m_1) \oplus \cdots \oplus F_k(m_\ell)$.

[5%] (b) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^{n/2}$, compute $t := F_k(\langle 1 \rangle \| m_1) \oplus \cdots \oplus F_k(\langle \ell \rangle \| m_\ell)$.

[5%] (c) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^{n/2}$, choose uniform $r \leftarrow \{0,1\}^n$, compute

$$t := F_k(r) \oplus F_k(\langle 1 \rangle \| m_1) \oplus \cdots \oplus F_k(\langle \ell \rangle \| m_\ell),$$

and let the tag be $\langle r, t \rangle$.

4.13 We explore what happens when the basic CBC-MAC construction is used with messages of different lengths.

[5%] (a) Say the sender and receiver do not agree on the message length in advance (and so $\mathsf{Vrfy}_k(m,t) = 1$ iff $t \stackrel{?}{=} \mathsf{Mac}_k(m)$, regardless of the length of m), but the sender is careful to only authenticate messages of length $2n$. Show that an adversary can forge a valid tag on a message of length $4n$.

[5%] (b) Say the receiver only accepts 3-block messages (so $\mathsf{Vrfy}_k(m,t) = 1$ only if m has length $3n$ and $t \stackrel{?}{=} \mathsf{Mac}_k(m)$), but the sender authenticates messages of any length a multiple of n. Show that an adversary can forge a valid tag on a new message.

4.14 Prove that the following modifications of basic CBC-MAC do not yield a secure MAC (even for fixed-length messages):

[5%] (a) Mac outputs all blocks $t_1, \ldots, t_\ell$, rather than just t_ℓ. (Verification only checks whether t_ℓ is correct.)

[5%] (b) A random initial block is used each time a message is authenticated. That is, choose uniform $t_0 \in \{0,1\}^n$, run basic CBC-MAC over the "message" $t_0, m_1, \ldots, m_\ell$, and output the tag $\langle t_0, t_\ell \rangle$. Verification is done in the natural way.