Problem Set 4

Due: Tuesday 12/06/2016, at the beginning of class.

1. (10 points) We said in class that (plain) CBC-MAC for fixed length messages is a secure MAC. Consider now a modification of CBC-MAC, where (similarly to CBC-encryption), all intermediate blocks $t_1, \ldots, t_\ell$ are output as part of the tag, rather than just $t_\ell$ (verification only checks that $t_\ell$ is correct).

Prove that this modified CBC-MAC is not secure (even for fixed-length messages).
Hint: you can find an attack for two-block messages, with a single query to the chosen message oracle.

2. (10 points) We showed a padding oracle attack on CBC-encryption (even for Authenticate-then-Encrypt, but here we just focus on the encryption aspect), which recovers the last block of the message, assuming the padding is according to the PKCS#5 standard (adding $N$ bytes, each encoding the number $N$).

Extend the attack to recover the entire message, not just the last block.
Hint: remember you can submit to your padding (decryption) oracle any ciphertext other than the challenge one, it doesn’t have to be the original challenge ciphertext with just one byte changed.

3. (15 points) Consider the following candidate $(\text{Gen, Mac, Vrfy})$ for arbitrary length MAC from a hash function $(\text{Gen}_H, H)$. $\text{Gen}$ starts by running $\text{Gen}_H$ to get $s$, which will be considered public and known to all, including the adversary (formally, we could add $s$ also as part of the output of $\text{Mac}_k$, but we will omit this here). It then selects a secret key $k \in \{0, 1\}^n$ uniformly at random. Define

$$\text{Mac}_s^k(m) = H^s(k \parallel m)$$

with the canonical verification algorithm.

As mentioned in class, this (with an unkeyed hash function) was a common choice in practice before HMAC.

(a) Prove that the above is a secure MAC if $H$ is modeled as a random oracle.

(b) Show that this is never a secure MAC when $H$ is constructed via the Merkle-Damgard transform using an underlying CRHF $h : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$.

4. (12 points) Let $(\text{Gen, H})$ be a CRHF. Define $\hat{\text{Gen}}$ to run $\text{Gen}$ twice independently, and output the resulting $(s_1, s_2)$. Now define $\hat{H}^{s_1, s_2}(x) = H^{s_1}(H^{s_2}(x))$.

Is $(\hat{\text{Gen}}, \hat{H})$ a CRHF? Prove your answer.

Does your answer change if $\hat{\text{Gen}}$ chooses $s_1 = s_2$ (namely $\hat{H}^s(x) = H^s(H^s(x))$)?

5. (10 points) Let $F$ be a PRF such that for $k \in \{0, 1\}^n$, we have $F_k : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$. Let $\text{Gen}(1^n)$ output a random $s \in \{0, 1\}^n$, and define $h^s(x) = F_s(x)$.

Is $(\text{Gen}, h)$ a fixed-length CRHF? Prove your answer.
6. **Not for Submission:** when we discussed authenticated encryption in class, a discussion with some students culminated in the following suggestion for an authenticated encryption, using a PRP with block size $n$. To encrypt a message $m \in \{0, 1\}^{n/3}$, choose a random $r \in \{0, 1\}^{n/3}$ and output $\text{Enc}_k(m) = F_k(0^{n/3} \parallel r \parallel m)$. Decryption is defined in the natural way, where $\text{Dec}_k(c)$ computes $F_k^{-1}(c)$, and if the first third of the bits is 0, outputs the last third of the bits, otherwise outputs $\bot$.

Prove that this is indeed an authenticated encryption (satisfying our definition of unforgeability and CCA security).

**Discussion:** In class I said that this is likely not CCA secure, since typically when you have to pad a message in a certain way to be valid, it opens to door to padding oracle attacks. I also said that a rule of thumb is that if your adversary may be active, you should always have your algorithms start by verifying a MAC, before proceeding to decrypt anything (for the same reason, to avoid padding oracle attacks). I was wrong about this particular scheme, and the reason (in terms of intuition and rules of thumb) is that this is an authenticated encryption scheme for fixed length (short enough to fit in one block with padding and randomness) messages. In this setting there isn’t a clear distinction regarding the first thing that we do – applying $F_k^{-1}$ – being a MAC verification or a decryption – it is (in some intuitive sense) doing both. For longer messages the mechanisms for encryption and for MAC, even when relying on PRPs, are different (see for example the differences between CBC-MAC and CBC-encryption), so this distinction makes more sense.