Ryan Young

CSE 3400

2/14/20

Homework 3

1.) *Show how an attacker can calculate $CRC\text{-}MAC_k(m)$ for a message m, given the value $CRC\text{-}MAC_k(m^l)$ for any message m^l such that* $|m^l| = |m|$.

Given one query the attacker should choose $q = m^l \oplus m$. The attacker knows that $CRC\text{-}MAC_k(m) = CRC\text{-}MAC_k(m^l) \oplus CRC\text{-}MAC_{0^\wedge|k|}(q) = CRC(k||m^l) \oplus CRC(0^{|k|}||q) = CRC(k||m^l) \oplus CRC(0^{|k|}||m \oplus m^l) = CRC(k||m) = CRC\text{-}MAC_k(m)$.

2.) *Hackme Inc. proposes the following highly efficient MAC, using two 64 bits keys k_1, k_2, for 64-bit blocks: $MAC_{k1,\,k2}(m) = ((m \oplus k_1)+k_2) \bmod 2^{64}$.* Show that this is not a secure MAC.

First, I would input $0^{|m|}$ into the MAC, this would result in, $((0^{|m|} \oplus k_1)+ k_2) \bmod 2^{64}$. Then I would input $1^{|m|}$ resulting in $((1^{|m|} \oplus k_1)+ k_2) \bmod 2^{64}$. Simplifying these two equations gives us: $(k_1 + k_2) \bmod 2^{64}$ and $(\sim k_1+k_2) \bmod 2^{64}$. Now because we have equations that have the inverse of k_1 and k_1 we can solve for k_2 namely by adding them together because $\sim k_1$ and k_1 should = 0. Then, once we have k_2 it is easy to compute k_1. I would input $0^{|m|}$ for the message m in order to not change k_1 and input the value of k_2 for k_2. Now the MAC is effectively broken completely because we were able to get both keys using only two queries.

3.) *Let $F:\{0,1\}^n \rightarrow \{0,1\}^l$ be a secure PRF, from n-bit strings to $l < n$-bit strings. Define $F':\{0,1\}^n \rightarrow \{0,1\}^l$ as: $F'_k(m) = F_k(m)||F_k(!m)$, i.e., concatenate the results of F_k applied to m and to the inverse of m. Present an efficient algorithm ADV F'_k which demonstrates that F' is not a secure MAC, i.e., outputs tuple(x,t) s.t. $x \in \{0,1\}^n$ and $t=F'_k(x)$. Algorithm ADV F'_k may provide input $m \in \{0,1\}^n$ and receive $F'_k(m)$, as long as x != m. You can present ADV F'_k by 'filling in the blanks' in the 'template' below, modifying and/or extending the template if desired, or simply write your own code if you like.*

An algorithm ADV should provide a query q = !x such that: $F^l_k(!x) = F_k(!x)||F_k(!(!x)$. This will allow ADV to take advantage of the fact that the PRF concatenates F_k(input message) with the inverse of F_k(input message). The last n/2 bits of the output of this query will give ADV the $F_k(x)$ and then you can concatenate this with the $F_k(!x)$ to get the tag for $F_k(x)$.

4.) *Let E be a secure (n + 1) bit block cipher and define the follow 2n-bit domain function:* $F_k(m_0||m_1) = E_k(0||m_0)||E_k(1||m_1)$. Show that F is not a secure 2 n-bit MAC.

F is a not a secure 2 n-bit MAC because if the adversary ADV were to make two queries $q_1 = F_k(m_0||0^n)$ and $q_2 = F_k(0^n || m_1)$, then ADV will be able to take the first n-bits of $F_k(m_0||0^n)$ and concatenate them with the last n-bits of $F_k(0^n || m_1)$.