Homework 24

Joe Baker, Brett Schreiber, Brian Knotten

March 20, 2018

42

Consider the example where $|m| = 2$ and $|k| = 1$. Therefore there are $2^2 = 4$ possible messages and $2^1 = 2$ possible keys. An eavesdropper Carol intercepts an encrypted message c. The encryption-decryption scheme (E, D) is public and so Carol knows it. But she doesn't know the key, so she tries all possible keys: 0 and 1. Carol runs $D_0(c)$ and gets m_0. Then she runs $D_1(c)$ and gets m_1. So Carol knows that the original message $m \in m_0, m_1$. So Carol can guess the message with probability $1/2$. However $|m| = 4$, meaning there are four possible messages: m_0, m_1, m_2, m_3. Therefore the probability of $E(m_2) = E(m_3) = 0$ and thus the distributions $E_{U_n}(m_0) \neq E_{U_n}(m_2)$.

More generally, let (E, D) be a scheme with message size m and key-size $n < m$. Let $m_0 \in M$, the set of all possible messages, let $k_0 \in K$, the set of all possible keys, and let $c_0 = E_{k_0}(m_0)$, the cipher text generated using k_0 when encrypting m_0. Then the probability of c being generated using any arbitrary key $k \in K$ when encrypting m_0 is at least the probability of any randomly chosen key being k_0 i.e. $P(E_k(m_0) = c) \geq P(k = k_0) > 0$.

Now consider the set of all possible decryptions of c_0 $D = \{D_k(c_0) | k \in K\}$. Clearly, $D \subseteq M$. Note that because the decryption function is well-defined, there must be at least as many keys as possible decryptions, so $|D| \leq |K|$. Then by our assumption, $|D| \leq |K| < |M|$, implying that $\exists m_1 \in M$ such that $m_1 \notin D$. Therefore the probability of c_0 being generated using any key $k \in K$ when encrypting m_1 is 0 i.e. $P(E_k(m_1) = c_0) = 0$. Thus, there exists a pair of messages m_0, m_1 such that $E_{U_n}(m_0) \neq E_{U_n}(m_1)$ and (E, D) is not a perfectly secret encryption scheme.

43

Assume that $f^k(x)$ is not a one-way permutation of x. $f^k(x)$ is still a permutation, since $f(x)$ is a permutation. So therefore $f^k(x)$ is not one-way. That means an algorithm A, given y and f^k will output the x such that $f^k(x) = y$ in polynomial time.

Construct an algorithm B as follows:
Given y, k and the one-way permutation f:
    Run A on f^k, y to get x such that $f^k(x) = y$.
    Repeat the following procedure $k - 1$ times:
        $x' := f(x)$
        $x := x'$
    Return x'

Since k is polynomial on n, then B is a polynomial algorithm, since it loops only $k - 1$ times. B returns the final value x' such that $f(x') = y$. Therefore B can reverse f. But f is a one way permutation. It cannot be cracked in polynomial time. There is a contradiction. Therefore f^k must also be a one-way permutation.

Alternative solution:

Let f be a one-way function and let $f^k(x) := f(f(...(f(x))))$ where f is applied k times. We will show that if f is a one-way function, then f^k is also a one-way function by contra-positive. Specifically we will show that if f^k is not a one-way function, then f is not a one-way function.

Assume f^k is not a one-way function. Then given $f^k(x) = y$, x can be computed in polynomial time. We also know that $f^k(x) = f(f^{k-1}(x')) = y$. Let $x := f^{k-1}(x')$. Since f^k is not a one-way function, given y we can determine x in polynomial time. This means given y, we can compute x such that $f(x) = y$ in polynomial time. Thus f is also not a one-way function.