Chapter 8 Homework 1-4, 6, 7

1. Let p be prime and let α be an integer with $p \not| \alpha$. Let $h(x) \equiv \alpha^x \pmod{p}$. Explain why $h(x)$ is not a good cryptographic hash function.

   h is not strongly collision-free because we can easily find $x_i, x_j \ni h(x_i) = h(x_j)$ if we know a message x_j. This can be accomplished by using Fermat's Little Theorem.
   Since $p \not| \alpha$, $\alpha^{p-1} \equiv 1 \pmod{p}$. So,

   $$h(x_i) \equiv \alpha^{x_i} \equiv \alpha^{x_i} \cdot \alpha^{p-1} \equiv \alpha^{x_i+p-1} \pmod{p}$$

   So, if we let $x_j = x_i + p - 1$, we get $h(x_i) = h(x_j)$ for $x_i \neq x_j$.

2. Let $h = pq$ be the product of two distinct large primes and let $h(x) = x^2 \pmod{n}$.

   (a) Why is h preimage resistant?
   In order to find x, we need $d \in \mathbb{Z} \ni (x^2)^d \equiv x \pmod{n}$.

   $$\Rightarrow x^{2d} \equiv x \pmod{n}$$
   $$\Rightarrow 2d \equiv 1 \pmod{\phi(n)}$$

   But, $(\phi(n), 2) \neq 1$ as $\phi(n) = (p-1)(q-1)$, which is the product of even integers. So, as we have seen, finding x is computationally infeasible.

   (b) Why is h not strongly collision-free?
   For a given x, let's consider $x_n \ni n \in \mathbb{Z}^*$.

   $$(x+n)^2 = x^2 + 2nx + n^2 \equiv x^2 \pmod{n}$$

   So, we have $x + n \neq n$ but $h(x) = h(x + n)$.

3. Suppose a message M is divided into blocks of length 160 bits:

   $$M_1||M_2||\ldots||M_l$$

   Let $h(x) = M_1 \oplus M_2 \oplus \ldots \oplus M_l$. Which of the properties $(1), (2), (3)$ for a hash function does h satisfy?

   (1) $\oplus$ is a relatively simple operation, so $h(x)$ <u>can</u> be calculated quickly.

   (2) preimage resistant: <u>No</u>
   Given $h(x)$, we can construct $M_1||M_2||\ldots||M_l$ in an arbitrary manner. Since we know $h(m)$, we have a 160 bit string of 0s and 1s. For each 0 in $h(M)$, we can assign an even number of 1s (or all 0s) to that position in the l blocks. For each 1 in $h(m)$, we can assign any odd number of 1s to that position in the l blocks.

   (3) Strongly collision-free: <u>No</u>
   We are only looking for $M_1 \neq M_2$ but $h(m_1) = h(m_2)$. We do not have to base this on a given $h(m)$. So, we can arbitrarily construct $M_1||M_2||\ldots||M_l$ for some number of blocks l. We could change an even number of 1s from a given position to 0, from 0 to 1, etc. and this changes the message but not the hash value.

4. In a family of 4, what is the probability no two people have birthdays in the same month?

$$P(E) = \left(1 - \frac{1}{12}\right)\left(1 - \frac{2}{12}\right)\left(1 - \frac{3}{12}\right) \approx 57.3\%$$

6. Suppose $f(x)$ is a function with n-bit outputs and with inputs much larger than n bits (this implies that collisions must exist). We know that, with a birthday attack, we have probability $\frac{1}{2}$ of finding a collision in $2^{\frac{n}{2}}$ steps.

   (a) Suppose we repeat the birthday attack until we find a collision. Show that the expected number of repetitions is

   $$1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 4 \cdot \frac{1}{16} + \ldots = 2$$

   Let

   $$S = 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 4 \cdot \frac{1}{16} + \ldots$$
   $$\frac{1}{2}S = 1 \cdot \frac{1}{4} + 2 \cdot \frac{1}{8} + 3 \cdot \frac{1}{16} + 4 \cdot \frac{1}{32} + \ldots$$

   So,

   $$S - \frac{1}{2}S = 1 \cdot \frac{1}{2} + (2-1) \cdot \frac{1}{4} + (3-2) \cdot \frac{1}{8} + \ldots = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots$$

   This is a finite geometric series with $a = 1$, $r = \frac{1}{2}$.

   $$\sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^k = \frac{1}{1 - \frac{1}{2}} = 2$$
   $$\Rightarrow \sum_{k=1}^{\infty} \left(\frac{1}{2}\right)^k = 2 - 1 = 1$$
   $$\Rightarrow \frac{1}{2}S = 1$$
   $$\Rightarrow S = 2$$

   (b) Assume that each evaluation of f takes time a constant times n. Show that the expected time to fin a collision for the function f is a constant times $n2^{\frac{n}{2}}$.

   Suppose each evaluation of f takes time c. It is expected that the birthday attack will require approximately $2^{\frac{n}{2}}$ steps to find a collision. If each of the n steps takes time c to be performed, the expected time is $cn2^{\frac{n}{2}}$.

   (c) Show that the expected time to produce a message $m_0, m'_0, \ldots, m_{t-1}, m'_{t-1}$ in § 8.5 is a constant times $tn2^{\frac{n}{2}}$.

   If each step takes come constant time cn and there are t components of the message, then the time to produce one round is ctn. Since we expect $2^{\frac{n}{2}}$ times to find a collision, the total time is $ctn2^{\frac{n}{2}}$.

7. Suppose we have an iterative hash function, as in § 8.5., but suppose we adjust the function slightly at each iteration. For concreteness, assume that the algorithm proceeds as follows: there is a compression function f that operates on inputs of a fixed length. There is also a function g that yields outputs of a fixed length, and there is a fixed initial value IV. The message is padded to obtain the desired format, then the following steps ae performed:

    1. Split the message M into blocks $M_1||M_2||\dots||M_l$.
    2. Let H_0 be the initial value IV.
    3. For $i = 1, 2, \dots, l$, let $H_i = f(H_{i-1}, M_i||g(i))$.
    4. Let $H(M) = H_l$.

    Show that the method of § 8.5 can be used to produce multicollisions for this hash function.

    This can be repeated by repeated applications of the birthday attack. In approximately $2^{\frac{n}{2}}$ steps, two blocks, say $M_{i_0}||g(i_0)$ can be found such that

    $$f(H_i, M_{i_0}||g(i_0)) = f(H_i, M_{i_1}g(i_1))$$

    This can be done for each of the values of t, so for each $M_{i_k}||g(i_k)$, $k \in \{0, 1\}$.
    If we then consider only M_{i_0} and M_{i_1}, for $i = 0, \dots, t-1$, we can construct 2^t different messages by choosing for each block M_i either M_{i_0} or M_{i_1} Each of the 2^t messages will have the same hash value and so we have a multicollision.