CO 487/687: Assignment #3 Due date: February 28 (Wednesday), 11:00 am

NOTE: You will be emailed a Crowdmark link for submitting the assignment on February 14. If you do not receive the link, please send an email to ajmeneze@uwaterloo.ca.

1. **Hash functions** (10 marks)

   (a) Is a preimage-resistant hash function necessarily second-preimage resistant? (Justify your answer.)

   Solution No. Suppose H is a preimage-resistant hash function $H : \{0,1\}^* \to \{0,1\}^n$. Define $H' : \{0,1\}^* \to \{0,1\}^n$ to be

   $$H'(x) = \begin{cases} H(x), & \text{if x is even} \\ H(x+1), & \text{if x is odd} \end{cases}$$

   Given any $x \in \{0,1\}^*$, if x is even, then $x-1$ is a second-preimage as $H'(x-1) = H(x-1+1) = H(x) = H'(x)$. If x is odd, then $x+1$ is a second-preimage. So H' is not second-preimaage resistant.
   Suppose H' is not preimage-resistant. Given a randomly chosen hash value $h \in_R \{0,1\}^n$ produced by H, with probability at least 0.5 h has a preimage which is even. If so we can find its preimage x' by invoking the preimage finding algorithm of H', namely x' s.t. $H'(x') = h$. Then $H(x') = H'(x') = h$. Since this 0.5 probability is not negligible, H is not preimage-resistant. By contradiction, H' is preimage-resistant.

   (b) Suppose that $H : \{0,1\}^* \to \{0,1\}^n$ is a collision-resistant hash function. Define a new hash function $G : \{0,1\}^* \to \{0,1\}^n$ by $G(x) = H(H(x))$. Prove that G is also collision resistant. (Note: As mentioned in class, such statements are best proven using the contrapositive statements. That is, you should prove that if G is not collision resistant, then H is not collision resistant.)

   Solution Suppose G is not collision resistant. There then exists a collision finding algorithm CF that outputs x, y s.t. $G(x) = G(y) = H(H(x)) = H(H(x))$. Compute $H(x), H(y)$. If $H(x) \neq H(y)$, then $H(x), H(y)$ is a collision since $H(H(x)) = H(H(y))$; Otherwise x, y is a collision. So there exists such a collision finding algorithm for H as well. Therefore G is collision-ressistant if H is collision-ressistant.

2. **Hash functions** (10 marks)
   Let H be an iterated hash function with compression function $f : \{0,1\}^{n+r} \to \{0,1\}^n$. (See slide 155 for a complete description of an iterated hash function.) Suppose that you are given an algorithm A which, on input $z \in \{0,1\}^n$, finds $s, t \in \{0,1\}^r$, $s \neq t$, such that $f(z,s) = f(z,t)$. Suppose that the running time of A is T. Let d be a positive integer, and let $D = 2^d$. Show how A can be used to find, in time approximately dT, D pairwise distinct messages $m_1, m_2, \ldots, m_D$ such that $H(m_1) = H(m_2) = \cdots = H(m_D)$.

   Solution Invoke A to find d distinct pairs $(s_i, t_i), i = 1, 2, \ldots, d$ such that $s_i \neq t_i, s_i \neq t_j$ if $i \neq j$ and $f(H_{i-1}, s_i) = f(H_{i-1}, t_i)$ where $H_{i-1} = f(H_{i-2}, s_{i-1}) = f(H_{i-2}, t_{i-1})$ for $i > 1$ and H_0 is an arbitrarily chosen n-bit binary string. Let k be a d-bit binary string and k_i is the ith bit of k. Construct string M_k by concatenating d s-bit binary strings where the ith block is s_i if k_i is 0 and t_i if k_i is 1. There are in total 2^d different messages as k varies from 0^d to 1^d. Since $f(H_0, M_{k1}) = f(H_0, M_{k'1}) = f(H_0, s_1) = f(H_0, t_1)$ and $f(H_{i-1}, M_{ki}) = f(H_{i-1}, M_{k'i}) = f(H_{i-1}, s_i) = f(H_{i-1}, t_i)$ where $k \neq k'$ and M_{ki} is the ith block of M_k, $H_d = f(H_{d-1}, M_{kd})$ is equal to $f(H_{d-1}, M_{kd})$. The length block b for both messages M_k and M'_k are of the same, so the final result $H(M_k) = f(H_d, b) = H(M'_k)$

3. **MAC schemes derived from hash functions** (10 marks)

Let $f : \{0,1\}^{256} \rightarrow \{0,1\}^{128}$ be a compression function. Let $H : \{0,1\}^* \rightarrow \{0,1\}^{128}$ be an iterated hash function derived from f as follows: To hash a message $m \in \{0,1\}^*$, do the following:

(i) Append a single 1 bit to the right of m, followed by just enough 0's so that the bitlength of the resulting message m' is a multiple of 128. Let the 128-bit blocks of m' be $m_1, m_2, \ldots, m_\ell$.

(ii) Let b be the 128-bit binary representation of ℓ.

(iii) Compute $H_1 = f(m_1, 0)$, $H_i = f(m_i, H_{i-1})$ for $2 \leq i \leq \ell$. Then $H(m) = f(b, H_\ell)$.

Consider now the following four MAC schemes each with 128-bit secret key k:

(a) $\mathrm{MAC}_k(m) = H(m) \oplus k$.

(b) $\mathrm{MAC}_k(m) = H(k, m) \oplus k$.

(c) $\mathrm{MAC}_k(m) = H(k, m) \oplus H(m)$.

(d) $\mathrm{MAC}_k(m) = H(m, k) \oplus k$.

Show that 3 of these 4 MAC schemes are insecure. That is, describe attacks on 3 of these 4 MAC schemes to demonstrate that they do not resist existential forgery by chosen-message attacks. (Note: It would help to to make your attacks as "practical" as possible—that is, use as few hash function evaluations as you can, and use as few calls to the MACing oracle as you can.)
(Note: As with all cryptographic systems considered in the course, you can assume you know all the details of the cryptographic system except for the secret key chosen by Alice and Bob. So, in this problem, you can assume that you know the complete description of f and H, but you do not know the secret key k.)

Solution (a) is insecure. Chose a message M and call MAC oracle once to get $c = H(m) \oplus k$. Compute $h = H(M)$ and recover the key $k = c \oplus h$. For any message m, $MAC_k(m) = H(m) \oplus k$ where k is now known.
(c) is insecure.

4. **Algorithm analysis** (10 marks)
   Describe a polynomial-time algorithm which, on input $n \in \mathbb{N}$, determines whether n is an integer fifth power; that is, your algorithm should determine if there exists an integer $a \in \mathbb{N}$ such that $n = a^5$. ($\mathbb{N}$ is the set of natural numbers.) Using big-O notation, give an upper bound on the number of bit operations your algorithm takes.

5. **RSA computations** (10 marks)
   Alice chooses $n = 1073$ and $e = 715$ for use in the basic RSA public-key encryption scheme.

   (a) What is Alice's private key?

   (b) Encrypt the message $m = 17$ for Alice.

   (The purpose of this problem is to make sure that you remember how to perform the basic RSA operations. You may use a calculator, but please show the main steps of your calculations. If you write a computer program, please hand in the program.)

You should make an effort to solve all the problems on your own. You are also welcome to collaborate on assignments with other students presently enrolled in CO 487/687. However, *solutions must be written up by yourself.* If you do collaborate, please *acknowledge your collaborators* in the write-up for each problem. *If you obtain a solution with help from a book, research paper, a web site, or elsewhere, please acknowledge your source.* You are *not* permitted to solicit help from online bulletin boards, chat groups, newsgroups, or solutions from previous offerings of the course.

The assignment should be submitted via Crowdmark before **11:00 am on February 28**. Late assignments will not be accepted except in *very* special circumstances (usually a documented illness of a serious nature). A high workload because of midterm tests and assignments in other courses will *not* qualify as a special circumstance.

Instructor and TA office hours:

Monday:	1:00 pm – 2:00 pm	Alessandra Graf (MC 5029)
	3:00 pm – 5:30 pm	Alfred Menezes (MC 5026)
Tuesday:	10:30 am – 11:30 am	Priya Soundararajan (MC 5466)
	11:30 am – 12:30 pm	Sam Jaques (QNC 4114)
	1:00 pm – 2:00 pm	Luis Ruiz-Lopez (MC 5486)
	3:00 pm – 4:00 pm	Elena Bakos Lang (MC 5474)
Thursday:	2:00 pm – 3:00 pm	Chris Leonardi (MC 5494)
Friday:	1:00 pm – 3:00 pm	Alfred Menezes (MC 5026)
