0368.3049.01 Introduction to Modern Cryptography Fall 2001

Assignment #2, due before class on Jan. 9, 2002

This assignment contains 4 "dry" problems and 2 "wet" ones. The answers to the latter should be given as the output of an XMAPLE session. Queries will be answered only if mailed till **Jan. 3**.

Problem 1: Cryptographic Hash Functions

Let $m = m_1 m_2 \ldots m_n$ where for every i $(i = 1, \ldots, n)$, m_i is a 128 bits binary string. Define a hash function, H to operate on messages of this form.

- h_0 is defined as the all zero string of length 128.
- for every i, $1 \leq i \leq n$, define $h_i = AES_{m_i}(h_{i-1})$.
- $H(m) = h_n$.

a. Show how to find collisions for H (namely two different messages that are mapped by H to the same string) using approximately 2^{64} AES applications.

b. Given a random string m, show how to find a different string m' such that $H(m) = H(m')$, using approximately 2^{64} AES applications.
Hint: Recall the attack on double DES.

Problem 2: Claw Free Permutations

Two permutations $f_0, f_1 : D \mapsto D$ is called *claw free* if it is infeasible to find $x, y \in D$ such that $f_0(x) = f_1(y)$.

a. Let p be a prime number, g a primitive element in Z_p, and $a \in Z_p^*$. Define the two permutations $f_0, f_1 : Z_p^* \mapsto Z_p^*$ by $f_0(x) = g^x \pmod p$ and $f_1(y) = ag^y \pmod p$. Assume it is infeasible to find a z such that $g^z = a$.
Prove that f_0, f_1 are claw free permutations.

b. Let $m = b_1 b_2 \ldots b_n$ be an n bit message (the b_i's are bits). Let f_0, f_1 be claw free permutations on D. Define the function H by

$$H(m) = f_{b_1}(f_{b_2} \ldots (f_{b_n}(IV) \ldots)) \ ,$$

where IV is the all zero string in D. For example, if $m = 011$ then $H(m) = f_0(f_1(f_1(IV)))$.
Assume that it is infeasible to find a $z \in D$ such that $f_0(z) = IV$ or $f_1(z) = IV$. Prove that H is a collision resistant hash functions. In other words, show that if $m_1 \neq m_2$ and $H(m_1) = H(m_2)$, then we can *efficiently* either find a pair $x, y \in D$ such that $f_0(x) = f_1(y)$, or a $z \in D$ such that $f_0(z) = IV$ or $f_1(z) = IV$. **Note that** m_1 and m_2 can have different lengths.

Problem 3: CBC MACs and variable length messages

In this problem we will explore the security of CBC MACs when the length of the message is allowed to vary. The constructions use a block cipher, $E : \{0,1\}^k \times \{0,1\}^n \mapsto \{0,1\}^n$ which you should assume to be secure ($E_K(t)$ is the encryption of n length t under k length key K).

In general, let $\mathbf{x} = x_1, x_2, \ldots, x_\ell$, where for each i, $x_i \in \{0,1\}^n$. For all the variants considered in this problem, the authentication of the message $\mathbf{x}$ is defined as the concatanation of $\mathbf{x}$ with $MAC_K(\mathbf{x})$, where K is the secret key (shared by Alice and Bob), and $MAC_K(\mathbf{x})$ is of length n.

We say that Fred, the forging adversary, succeeds if after seeing a small number of messages $\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_s}$ *of his choice* and their MACs under the *unknown* secret key K, he can produce a new message $\mathbf{w}$ ($\mathbf{w} \notin \{\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_s}\}$) together with $MAC_K(\mathbf{w})$. We emphasize that w can, and typically will, be constructed out of pieces depending on the z_i's. By small number we mean s is either a constant or at most a (fixed) polynomial in n, the block length of $\mathbf{x}$. In addition to the number s of message/MAC pairs, Fred is also limited to polynomial time computations (polynomial in n).

Remark: This type of forgery is called *adaptive existential forgery* (adaptive since the choice of each z_{i+1} can depend on all previous i message/MAC pairs, and existential because it demonstrates the existence of a message whose MAC can be forged). This is the strongest form of "reasonable adversary" considered in the crypto world.

a. Consider the application of "regular" CBC MAC to messages of arbitrary length. Formally, given $\mathbf{x} = x_1, x_2, \ldots, x_\ell$, we define $y_0 = 0^n$, and for $0 \leq i \leq \ell - 1$, $y_{i+1} = E_K(y_i \bigoplus x_{i+1})$. Then $CBC - MAC_K(\mathbf{x}) = y_\ell$. Show that this MAC is completely insecure: Break it with a constant number of queries.

b. In order to overcome the problem of applying "regular" CBC MAC to messages of arbitrary length, consider the following patch:

$$MAC_K(x_1, x_2, \ldots, x_\ell) = CBC - MAC_K(x_1, x_2, \ldots, x_\ell, \ell) \ ,$$

where ℓ, the number of blocks in $\mathbf{x}$ is written in binary using n bits. Show that this patch does not hold water either: Break it with a constant number of queries.

c. Consider the following attempt to allow one to MAC messages of arbitrary length. The domain for the MAC is $(\{0,1\}^n)^+$. To MAC the message $\mathbf{x} = x_1, x_2, \ldots, x_\ell$, under the secret key (K, K'), compute $CBC - MAC_K(\mathbf{x}) \bigoplus K'$, where K has k bits and K' has n bits. Show that this MAC is completely insecure: Break it with a constant number of queries.

Problem 4: Orders

a. Let a, m be two positive integers, with $1 \leq a \leq m - 1$. The order of a modulo m, $ord_m a$, is defined as the minimum positive integer ℓ such that $a^\ell = 1 \pmod{m}$, and ∞ if no such ℓ exists. Prove that $ord_m a < \infty$ if and only if $\gcd(a, m) = 1$.

b. Let x be an integer, and let p be an odd prime divisor of $x^{16} + 1$. Prove that $p = 1 \pmod{32}$.

Problem 5: Primitive Elements in Z_p

a. Let $p > 2$ be a prime number and let g be a primitive element in Z_p^*. Find a characterization of all integers e, $1 \leq e \leq p - 2$ such that $g^e \bmod p$ is also a primitive element in Z_p^*. Prove the characterization.

b. Find the two *largest* prime numbers that are *smaller* than $2^{127} - 1$, using Maple's `isprime(m)` for efficient primality testing. Let p_1 and p_2 denote these two primes. Print p_1 and p_2 in a compact manner (not the 40 digit representations). Let p_0 denote $2^{127} - 1$ (verify for yourself that it is indeed a prime number). Find and print the factorizations of $p_i - 1$ for $i = 0, 1, 2$. Does any of the $p_i - 1$ have a prime factor larger than $\sqrt{p_i - 1}$? Take $p = p_i$ with the largest factor of the three. What is the largest power of 2 smaller than the largest factor of $p - 1$?

c. For p of section (b), find at random an integer g, $g > 10^7$, such that g is a primitive element of Z_p but $g + 1$ is *not* a primitive element of Z_p. Print a Maple session that explicitly proves both statements.

Hint: You can make the search easier by using Maple's commands

$$FF := \texttt{GF}(p, 1);$$

$$x := FF[\texttt{ConvertIn}](g);$$

$$FF[\texttt{isPrimitiveElement}](x);$$

for verifying g's "primitivity status" in the field $Z_p = GF(p, 1)$. However in your "explicit proof" you should *not* use such `FF` implementation, but rather use only the `mod` p function and &^ (exponentiation) to appropriate powers.

Problem 6: Primality Testing

a. Let $N = (2^{127} - 1) * (2^{127} - 801)$. Clearly N is composite, and we have just given you a short proof of this fact. Come up with the simplest proof you can think of for N being composite, which *does not* use the factorization. You are encouraged to use Maple, but using Maple's `isprime(N)` is obviously not acceptable.

b. Find the *smallest* prime number p that is *larger* than 2^{127}. Run the Miller-Rabin primality test on p with three independent random integers a_1, a_2, a_3. Submit all relevant equalities.

c. For every odd integers N in the range $2^{127} < N < p$, supply the shortest "explicit proofs" you can think of for N being composite.