MATH/CMSC 456 (Washington) Exam 2 May 7, 2003

Do each problem on a separate piece of paper. That is, do problem 1 on page 1, problem 2 on page 2, etc.

1. (20 points=6+7+7) (a) Suppose you know that

$$3^6 \equiv 44 \pmod{137}, \quad 3^{10} \equiv 2 \pmod{137}.$$

Find a value of x with $0 \le x \le 135$ such that $3^x \equiv 11 \pmod{137}$.
(b) Let E be the elliptic curve $y^2 \equiv x^3 + 1 \pmod 7$. Evaluate the sum $(0,1)+(3,0)$ on E.
(c) Let H be the function that takes as input a large integer and reduces it mod 10^{100}. That is, $H(x) = x \pmod{10^{100}}$. Why is H not a good cryptographic hash function?

2. (20 points = 4+8+8) An attack on discrete logarithms goes as follows. Let p be a prime and let α be a primitive root mod p. Given β, we want to find x such that $\beta \equiv \alpha^x \pmod p$. Make two lists of length M (for the value of M, see part (a)). The first list is $\beta\alpha^{-i} \pmod p$ for M random values of i. The second list is α^j for M random values of j.
(a) Suppose there is a match between an element of the first list and one on the second list. Show that this yields a value of x that solves the discrete log problem.
(b) Suppose p is approximately 10^{30}. Approximately how large should M be so that there is a 50% chance of a match? (Your answer should be something like 10^{10} or 10^{19}. You don't need to be more accurate than a power of 10.)
(c) Describe the analog of this procedure for elliptic curves. Namely, let E be an elliptic curve with N points and suppose A and B are points on E. Suppose $B = xA$ for some integer x. Give the procedure for finding x.

3. (20 points: 10+10) Let p be a large prime and let α be a primitive root mod p. Let f be a function that maps integers to integers (so $f(x)$ is an integer when x is an integer). Alice wants to sign a message m, which is represented as an integer mod p. Alice chooses a secret integer a and computes $\beta \equiv \alpha^a \pmod p$. She makes p, f, α, β public and keeps a secret. To sign m, Alice does the following:

(1) She chooses a secret random integer k with $\gcd(k, p-1) = 1$.
(2) She computes $r \equiv m\alpha^{-k} \pmod p$.
(3) She computes $s \equiv k^{-1}(1 - f(r)a) \pmod{p-1}$.
(4) The signed message is (m, r, s).

Bob verifies the signature as follows:

(1) He computes $v_1 \equiv \alpha r^s \beta^{-f(r)} \pmod p$.
(2) He computes $v_2 \equiv m^s \pmod p$.
(3) He declares the signature valid if $v_1 \equiv v_2 \pmod p$.

(a) Show that if Alice performs the required steps correctly, then $v_1 \equiv v_2 \pmod p$.

(b) Suppose Alice uses the constant function satisfying $f(x) = 0$ for all x. Let m be Eve's message. Show how Eve can forge Alice's signature on m (that is, describe how Eve can produce a signed message (m, r, s) that Bob will declare to be valid).

4. (20 points=10+10) Suppose p is a large prime and α is a primitive root mod p. Suppose $\beta \equiv \alpha^x \pmod{p}$ for some x. Peggy wants to prove to Victor that she knows x without telling Victor the value of x. (Victor knows p, α, β.) They do the following:

(1) Peggy chooses three random integers r_1, r_2, r_3 such that $r_1 + r_2 + r_3 \equiv x \pmod{p-1}$.
(2) Peggy computes $m_i \equiv \alpha^{r_i} \pmod{p}$ for $i = 1, 2, 3$.
(3) Peggy sends m_1, m_2, m_3 to Victor.
(4) Victor checks that $m_1 m_2 m_3 \equiv \beta \pmod{p}$.
(5) Victor chooses two integers $j, k \in \{1, 2, 3\}$ and sends them to Peggy.
(6) Peggy sends r_j and r_k to Victor.
(7) Victor checks that $v_j \equiv \alpha^{r_j} \pmod{p}$ and that $v_k \equiv \alpha^{r_k} \pmod{p}$.

(a) Suppose Peggy does not know x but guesses correctly that Victor will ask for $j = 1$ and $k = 3$. Describe what Peggy should do so that Victor does not find out that she doesn't know x.
(b) Suppose that Peggy does not know x. Why is it likely, after several repetitions of the above procedure, that Victor will discover that Peggy does not know x?

5. (20 points) Consider the following Feistel cryptosystem consisting of two rounds. The key K is the same for each round and has 64 bits. The input for the ith round consists of 64 bits, divided into a left half and a right half: $L_{i-1}R_{i-1}$, where L_{i-1} and R_{i-1} each have 32 bits. The output is $L_i R_i$, where $L_i = R_{i-1}$ and $R_i = L_{i-1} \oplus f(K, R_{i-1})$. The function f is given by $f(K, R) \equiv R \oplus R^K \pmod{2^{64}}$, written as a 64-bit string.

If you receive the ciphertext $L_2 R_2$, describe how you can use the encryption algorithm to decrypt it and obtain $L_0 R_0$. Show that this decryption works. (You may not simply quote results about this type of decryption.)