University of Waterloo
Department of C&O

PhD Comprehensive Examination in Cryptography
Spring 2002
Examiners: A. Menezes, D. Stinson and E. Teske

June 27, 2002
9:00 am — 12:00 pm
MC 5158A

Instructions

Answer any *six* of the seven questions.

Questions

1. **Attempts to strengthen DES against exhaustive key search attacks**
   Recall that DES is a symmetric-key encryption scheme with a 56-bit key, and 64-bit plaintext and ciphertext blocks. Consider the following proposal for a new symmetric-key encryption scheme based on DES. The secret key for the new scheme is $k = (k_1, k_2)$, where $k_1 \in \{0,1\}^{56}$ and $k_2 \in \{0,1\}^{64}$ (so k is a 120-bit key). Let $m \in \{0,1\}^{64}$ be a plaintext message. Then encryption is defined as follows:

   $$E_k(m) = \mathrm{DES}_{k_1}(m \oplus k_2).$$

   (a) Show how this encryption scheme can be totally broken—that is, the secret key k can be recovered—by a known-plaintext attack using *roughly* 2^{56} DES encryption/decryption operations. Your attack should have little space requirements. You may assume that you have a moderate number of plaintext-ciphertext pairs $(m_i, c_i = E_k(m_i))$. Briefly justify why the number of such pairs you use is sufficient to uniquely determine the key with high probability.

   (b) Is the encryption scheme with encryption function $E_k(m) = \mathrm{DES}_{k_1}(m) \oplus k_2$ any more secure than DES? [Briefly justify your answer.]

   (c) Is the encryption scheme with encryption function $E_k(m) = \mathrm{DES}_{k_1}(m \oplus k_2) \oplus k_3$ any more secure than DES? (Here, $k = (k_1, k_2, k_3)$ where $k_1 \in \{0,1\}^{56}$ and $k_2, k_3 \in \{0,1\}^{64}$.) [Briefly justify your answer.]

2. **Hash Functions**
   Suppose $h_1 : \{0,1\}^{2m} \longrightarrow \{0,1\}^m$ is a collision resistant hash function.

   (a) Define $h_2 : \{0,1\}^{4m} \longrightarrow \{0,1\}^m$ as follows:

   1. Write $x \in \{0,1\}^{4m}$ as $x = x_1 \parallel x_2$, where $x_1, x_2 \in \{0,1\}^{2m}$.
   2. Define $h_2(x) = h_1(h_1(x_1) \parallel h_1(x_2))$.

   Prove that h_2 is collision resistant.

   (b) For an integer $i \geq 2$, define a hash function $h_i : \{0,1\}^{2^i m} \longrightarrow \{0,1\}^m$ recursively from h_{i-1}, as follows:

   1. Write $x \in \{0,1\}^{2^i m}$ as $x = x_1 \parallel x_2$, where $x_1, x_2 \in \{0,1\}^{2^{i-1} m}$.
   2. Define $h_i(x) = h_1(h_{i-1}(x_1) \parallel h_{i-1}(x_2))$.

   Prove that h_i is collision resistant.

3. **Carmichael numbers**
   Assume throughout this question that n is square-free (i.e., n is not divisible by the square of a prime). Then $n = p_1 \cdots p_\ell$, where $p_1, \ldots, p_\ell$ are distinct primes. Such an integer n is a *Carmichael number* if $a^{n-1} \equiv 1 \pmod{n}$ for all integers a that are relatively prime to n.
   You may use the following facts in your solutions: (i) $a^{n-1} \equiv 1 \pmod{n}$ if and only if $a^{n-1} \equiv 1 \pmod{p_i}$ for all i, $1 \leq i \leq \ell$; (ii) For every prime number p, there exists a primitive element mod p.

   (a) Suppose that n is a Carmichael number. Prove that $p_i - 1$ divides $n - 1$ for all i, $1 \leq i \leq \ell$.

(b) Suppose that $p_i - 1$ divides $n-1$ for all i, $1 \le i \le \ell$. Then prove that n is a Carmichael number.

(c) Prove that 561 is a Carmichael number.

4. **Speeding up RSA decryption**
Let (n, e) be Alice's RSA public key, and let d be her corresponding private key. Recall that the RSA encryption operation is $c = m^e \bmod n$, while the RSA decryption operation is $m = c^d \bmod n$. One way to speed up RSA decryption is to precompute $d_p = d \bmod (p-1)$ and $d_q = d \bmod (q-1)$. Then decryption of a ciphertext c can be performed by computing $m_p = c^{d_p} \bmod p$ and $m_q = c^{d_q} \bmod q$, and then finding m, $0 \le m \le n-1$, such that

$$m \equiv m_p \pmod{p}$$
$$m \equiv m_q \pmod{q}.$$

(a) Describe a procedure (i.e., a formula) that Alice can use to compute m efficiently, given m_p and m_q.

(b) Prove that m is the correct decryption of c. That is, prove that $m \equiv c^d \pmod{n}$.

(c) Briefly justify the assertion that this method of decryption can speed up RSA decryption by approximately 75%, given that a modular exponentiation operation modulo n can be done in $O(\log n)^3$ bit operations and given that $p \approx q$.

(d) Devise an algorithm which, on input n and e, factors n in $O(\min(d_p, d_q))$ steps. (A "step" is any operation whose running time is polynomial in $\log n$.) State any assumptions you may make. [This exercise shows that Alice should not try to speed up decryption by selecting d so that d_p and d_q are too small.]

5. **Bit security of the Discrete Logarithm Problem**
Let p be a prime with $p \equiv 3 \pmod{4}$. Let $\alpha \in \mathbb{Z}_p$ be a generator of $\mathbb{Z}_p^*$. The discrete logarithm problem in $\mathbb{Z}_p^*$ is the following: given α and $\beta \in_R \mathbb{Z}_p^*$, find the integer a, $0 \le a \le p-2$, such that $\beta \equiv \alpha^a \pmod{p}$.
Let $L_1(\beta)$ denote the least significant bit of a. That is, $L_1(\beta) = 0$ if a is even, and $L_1(\beta) = 1$ if a is odd.
Let $L_2(\beta)$ denote the second least significant bit of a. That is, $L_2(\beta) = 0$ if $a \equiv 0$ or $1 \pmod{4}$, and $L_2(\beta) = 1$ if $a \equiv 2$ or $3 \pmod{4}$.

(a) Let $\gamma \in \mathbb{Z}_p^*$ be a quadratic residue modulo p. Show that the two square roots of γ modulo p are $\pm\gamma^{(p+1)/4}$.

(b) Show that $L_1(\beta)$ can be efficiently computed given p, α, β.

(c) Prove that $L_1(\beta) \ne L_1(-\beta)$.

(d) Suppose that you have an efficient algorithm A (an oracle) for computing $L_2(\beta)$ given p, α, β. Devise an efficient algorithm for solving the discrete logarithm problem. Briefly justify that your algorithm is *correct* and *efficient*.

6. **Hash Functions and DSA**
We recall the DSA signature scheme. The system parameters consist of a 1024-bit prime p, a 160-bit prime divisor q of $p-1$, and an element $g \in \mathbb{Z}_p^*$ of order q. SHA-1 is a 160-bit hash function. Alice's private key is $a \in_R [0, q-1]$, while her public key is $h = g^a \bmod p$. To sign a message $M \in \{0,1\}^*$, Alice does the following:

(i) Select $k \in_R [1, q-1]$.
(ii) Compute $m = \text{SHA-1}(M)$.
(iii) Compute $r = (g^k \bmod p) \bmod q$, and check that $r \neq 0$.
(iv) Compute $s = k^{-1}\{m + ar\} \bmod q$, and check that $s \neq 0$.
(v) Alice's signature on M is (r, s).

To verify A's signature (r, s) on M, Bob does the following:

(i) Obtain an authentic copy of Alice's public key h.
(ii) Compute $m = \text{SHA-1}(M)$.
(iii) Check that $1 \leq r, s \leq q-1$.
(iv) Compute $u_1 = ms^{-1} \bmod q$ and $u_2 = rs^{-1} \bmod q$.
(v) Accept iff $r = (g^{u_1} h^{u_2} \bmod p) \bmod q$.

Recall that a signature scheme is *secure* if it is existentially unforgeable by chosen-message attacks. It is *insecure* if it is not secure.

(a) Define what it means for SHA-1 to be preimage resistant.

(b) Define what it means for SHA-1 to be 2nd preimage resistant.

(c) Define what it means for SHA-1 to be collision resistant.

(d) Prove that DSA is insecure if SHA-1 is not preimage resistant.

(e) Prove that DSA is insecure if SHA-1 is not 2nd preimage resistant.

(f) Prove that DSA is insecure if SHA-1 is not collision resistant.

7. **Duplicate Signatures**
The Elliptic Curve Digital Signature Algorithm (ECDSA) is as follows: Let p be a prime, and let E be an elliptic curve defined over F_p. Let A be a point on E having prime order q, such that the Discrete Logarithm problem in $\langle A \rangle$ is infeasible. Let $\mathcal{P} = \{0,1\}^*$, $\mathcal{A} = Z_q{}^* \times Z_q{}^*$, and define

$$\mathcal{K} = \{(p, q, E, A, m, B) : B = mA\},$$

where $0 \leq m \leq q-1$. The values p, q, E, A and B are the public key, and m is the private key. The signature for a message $x \in \mathcal{P}$ is computed as follows: For $K = (p, q, E, A, m, B)$, and for a (secret) random number k, $1 \leq k \leq q-1$, define

$$sig_K(x, k) = (r, s),$$

where

$$\begin{aligned} kA &= (u, v) \\ r &= u \bmod q, \quad \text{and} \\ s &= k^{-1}(\text{SHA-1}(x) + mr) \bmod q. \end{aligned}$$

(If either $r = 0$ or $s = 0$, a new random value of k should be chosen.)

(a) Suppose that x and x' are any two messages. Suppose that x is signed using random number k and x' is signed with random number $k' = -k \bmod q$. Prove that $sig_K(x, k) = sig_K(x', k')$ if and only if $\text{SHA-1}(x) + \text{SHA-1}(x') + 2mr \equiv 0 \pmod q$.

(b) Two messages x and x' are said to have *duplicate ECDSA signatures* if SHA-1$(x) \neq$ SHA-1(x') but $sig_K(x, k) = sig_K(x', k')$ for some integers k, k'. Suppose that the public key parameters p, q, E and A are fixed. Given any two messages x and x' such that SHA-1$(x) \neq$ SHA-1(x'), show that it is possible for Alice to choose a private key m (and hence a corresponding public key B) so that x and x' have duplicate ECDSA signatures under the key p, q, E, A, m and B.

(c) Suppose that Alice, say, signs message x with signature (r, s), and then later claims that she really signed the message x', where (r, s) is also a signature on x'. Show that an adversary can now easily compute Alice's secret key.

(d) We have shown that Alice can choose her private key in such a way that she can later construct duplicate signatures on two messages x and x'. Does this property mean that the ECDSA is "insecure"? (Discuss)