Test 2. Due 04/12/04 4:00pm

This test is take home. It must be an individual work. You may, however, consult your textbook and notes. Any other form of getting help, such as consulting other textbooks or web sites, is a violation of the honor code. Providing information to others that leads to solutions is also a violation of the honor code.

Show your work and justify all your answers. You will be graded not only on the correctness of your answer, but also on the clarity you express it.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100</strong></td>
<td></td>
</tr>
</tbody>
</table>

I have abided by the Academic Honor Code on this test.

Name: __________________________ Signature: __________________________ Date: __________________________

1. (20 points) Suppose a KDC infrastructure is used to authenticate and distribute users’ public keys. Alice and Bob do not share secrets, and they do not know each other’s public keys. Suppose Alice and Bob want to use AES to secure their communications over the Internet. To reduce bottleneck traffic to KDC, devise a secure key distribution scheme using the existing KDC infrastructure and technology for the following purpose: Alice and Bob only need to use KDC once in a period of time (that is, if Bob has already contacted KDC, then Alice doesn’t need to), and can generate a session key for each communication Alice and Bob may have in that period without contacting KDC again.

2. (20 points) Bob suspects that his private key $d$ for using RSA has been compromised. Rather than generating a new modulus, he decides to generate a new $d$ and $e$ with the old modulus $n$. Is this safe? Defend your answer.

3. (20 points) To have a secure implementation of the RSA public-key encryption algorithm, the modulus $n$ should be a product of two large prime numbers $p$ and $q$, which should have approximately the same number of binary digits. Let $\ell$ denote the number of binary digits in $n$. We can use a fast squaring algorithm to calculate $M^e(\text{mod } n)$ and

Dr. J. Wang
$C^d \pmod{n}$. Since $M < n$, $M \pmod{n} = M$, and so when $M$ (similarly when $C$) is large (e.g., when $M$ contains $\ell$ binary digits), a naive algorithm that calculates $M^2 \pmod{n}$ still needs to deal with multiplications of large numbers, resulting in a larger number of length about $2\ell$ before modulus is applied to reduce its length. Device an algorithm to calculate $M^2 \pmod{n}$ (similarly $C^2 \pmod{n}$) so that the largest number ever occurs during the calculation will only have at most $3\ell/2$ binary digits.

4. (20 points) Consider the following public-key scheme based on discrete logarithms, which is closely related to the Diffie-Hellman technique. This scheme uses two global elements: a prime number $q$ and a primitive root, $a$, of $q$. User A selects a private key $X_A$ and calculates a public key $Y_A = a^{X_A} \pmod{q}$. User B does the same: Select $X_B$ and calculate $Y_B = a^{X_B} \pmod{q}$.

User A encrypts a plaintext $M < q$ intended for user B as follows:

(a) Choose a random integer $k$ such that $1 \leq k \leq q - 1$.
(b) Compute $K = (Y_B)^k \pmod{q}$.
(c) Encrypt $M$ as the pair of integers $(C_1, C_2)$, where $C_1 = a^K \pmod{q}$ and $C_2 = KM \pmod{q}$.

User B recovers the plaintext by calculating $K$ using $C_1$ and $X_B$, which allows B to obtain $M$ using $C_2$. Explain how and defend your answer.

5. (20 points) Suppose $a_1a_2a_3a_4$ is a 4-byte word. Each $a_i$ can be viewed as an integer in the range 0 to 255, represented in binary. In a big-endian architecture, this word represents the integer

$$a_42^{24} + a_32^{16} + a_22^8 + a_1.$$

In a little-endian architecture, this word represents the integer

$$a_42^{24} + a_32^{16} + a_22^8 + a_1.$$

MD5 assumes a little-endian architecture. Show how the MD5 addition operation $X + Y$ would be carried out on a big-endian machine.