T-79.159 Cryptography and Data Security **Spring 2003**
Tutorial 6: Solutions
18.–19.3.2003

1. Let $n = pq$ be the modulus, and let (e_1, d_1) and (e_2, d_2) be the two exponent pairs. Assume that $\gcd(e_1, e_2) = 1$ (this holds with a fairly high probability). Suppose that the same (unknown) plaintext x have been encrypted using the two keys. Let $y_1 \equiv x^{e_1} \pmod n$ and $y_2 \equiv x^{e_2} \pmod n$ be the corresponding ciphertexts. Since $\gcd(e_1, e_2) = 1$, the Euclidean algorithm can be used to express $1 = ae_1 + be_2$ for some integers a, b. Now, $y_1^a y_2^b \equiv (x^{e_1})^a (x^{e_2})^b \equiv x^{ae_1+be_2} \equiv x^1 \equiv x \pmod n$. That is, the plaintext can easily be recovered.

2. Recall that the Chinese remainder theorem states that the system of equations

$$\begin{aligned} z &\equiv a_1 \pmod{m_1} \\ z &\equiv a_2 \pmod{m_2} \\ &\vdots \\ z &\equiv a_k \pmod{m_k} \end{aligned}$$

   with $i \neq j \Rightarrow \gcd(m_i, m_j) = 1$ has a unique solution modulo $M = m_1 m_2 \cdots m_k$. The solution can be computed by setting $M_i = M/m_i$ and $b_i = a_i M_i (M_i^{-1} \bmod m_i)$. The solution is $z = b_1 + b_2 + \cdots b_k$.

   Since most of the computation time is spent computing the exponentiations $y_p = x^d \bmod p$ and $y_q = x^d \bmod q$, it is likely that the random error occurs during one of the exponentiations. Assume that the error occurs during the exponentiation (say) modulo p. This gives the faulty result $y'_p \not\equiv y_p \bmod p$. Applying the Chinese remainder theorem gives the faulty signature $y' \not\equiv y \mod n$.

   Note that that $y' \equiv y'_p \not\equiv y_p \pmod p$ but $y' \equiv y_q \pmod q$. Thus, ${y'}^e - x \equiv {y'_p}^e - x \not\equiv y_p^e - x \equiv x^{de} - x \equiv 0 \pmod p$, but ${y'}^e - x \equiv y_q^e - x \equiv x^{de} - x \equiv 0 \pmod q$. That is, p does not divide ${y'}^e - x$, but q divides ${y'}^e - x$. It follows that $\gcd({y'}^e - x, n) = q$. Since the message x and the faulty signature y' are known, this can easily be computed.

   That is, based on the known message x and the faulty signature y', it is trivial to extract the private key. This illustrates the importance of either preventing errors (caused by an attacker) or detecting them and handling them correctly in implementations.

3. The problem with the protocol is that identifications can be forwarded. Suppose A wants to identify herself to M. Then M can easily idenify himself as A to B as follows.

   - A picks a random nonce N_A and sends $a = E_M(A, N_A)$ to M.
   - M decrypts a and sends $a' = E_B(A, N_A)$ to B.
   - B decrypts a', picks a random nonce N_B and sends $c = E_A(N_A, N_B)$ to M.

- M forwards c to A.
- A decrypts c and sends $z = E_M(N_B)$ to M.
- M decrypts z and sends $z' = E_B(N_B)$ to B.
- B verifies that $E_B^{-1}(z') = N_B$.

A simple way to prevent this particular attack is to include the names (A, B) in all the messages.

4. (a) If each step is correctly computed, $g^z = g^{cw+r} = g^r(g^w)^c = ah^c \pmod p$.

   (b) The group operation in the first and last steps is multiplication modulo p. Since the responce z computed in the third step is used as an exponent in the verification equation and the order of g is q, the exponent is reduced modulo q, not modulo p. If x is any integer and $y = x \bmod q$, $x = y + kq$ for some k and $g^x = g^y(g^q)^k = g^y \pmod p$, since $g^q = 1 \pmod p$.

   Since $q|(p-1)$ it would also work to reduce z modulo $p-1$.

   (c) Since B only sends a random challenge $c \in \{0,1\}^t$ to A, there is no way for A to identify B. If mutual identification is needed, the protocol can be repeated with swapped roles (and different values of w and h).

   (d) Suppose an attacker can impersonate A with a "farily high" probability. Then the attacker knows an a such that it can find an accepting responce z for a significant fraction of the challenges c. In particular this means that the attacker can find two accepting conversations (a, c, z) and (a, c', z') with a non-neglecatble probability, where $c \neq c'$.

   Let $a = g^r$ for some (unknown) r. The conversations (a, c, z) and (a, c', z') are accepting if and only if $z = cw + r \pmod q$ and $z' = c'w + r \pmod q$. Since $z - z' = (cw + r) - (c'w + r) = (c - c')w \pmod q$ and $c \neq c'$, w can be computed as $w = (z - z')(c - c')^{-1} \pmod q$.

   That is, an attacker that can impersonate A with a "fairly high" probability is also able to compute w from $h = g^w$.