University of Haifa Spring Semester – 3/5/11

Introduction to Cryptography – Exercise no. 4

Submit in Pairs/Single to **mailbox** 19 by 25/5/11, 1:00 p.m.

1. Prof. Namlleh observed that in the Diffie-Hellman key exchange protocol, each user U publishes $Y_U = g^{X_U} \bmod p$, where p is a known large prime number, and $g \in Z_p$ is a **generator** of Z_p.

   (a) Show that given p, g, $Y_U = g^{X_U} \bmod p$ it is possible to reveal the least significant bit of X_U.

   His assistant, Dr. Eiffid, proposed the following algorithm to solve the DLOG problem, given the least significant bit of X_U:

   - Compute the least significant bit of X_U.
   - If the LSB is 1, let $Y'_U = Y_U \cdot g^{-1}$, otherwise let $Y'_U = Y_U$.
   - Compute the square root of Y'_U, and repeat the algorithm on the square root until $Y'_U = 1$.
   - By collecting the LSBs one can reconstruct X'.

   (b) Show that each Y'_U is a quadratic residue modulo p.

   (c) Show that the LSBs collected by the algorithm, denoted by X', satisfies $g^{X'} \equiv Y_U \bmod p$. Conclude that the secret key of U is $X_U \equiv X' \bmod p-1$.

   (d) Explain why the above algorithm cannot compute the DLOG X_U in spite of the above.

2. (a) Calculate the Jacobi symbol $\left(\frac{1000}{78921}\right)$ using the methods taught in class, describe your calculations.

   (b) Prove that if $n = p_1^{e_1} \cdot p_2^{e_2} \cdots p_\ell^{e_\ell}$ then

   $$\varphi(n) = n \cdot \prod_{i=1}^{\ell} \left(1 - \frac{1}{p_i}\right)$$

   (c) For a cipher with a key K a message M is called a fixpoint if $E_K(M) = M$. Prove that in an RSA system with $n = pq$ and a public key (n, e) the number of fixpoints is $(gcd(p-1, e-1)+1) \cdot (gcd(q-1, e-1)+1)$.

3. (a) Find the modular square roots of 7 modulo 31, using the algorithm we saw in class. Describe your calculations.

   (b) Find the modular square roots of 2 modulo 17, using the algorithm we saw in class. Describe your calculations.

   (c) Find the modular square roots of 53 modulo 143, using the algorithm we saw in class. Describe your calculations.

4. This question discuss RSA in the real world.

   (a) Alice has an RSA public key of the form $(n, e = 3)$. When Bob wants to send Alice a message, he encrypts it using Alice's public key (i.e., computes $c = m^3 \bmod n$). If for some reason Alice fails to receive correctly the message, she informs Bob on the matter, and he retries to send the message again. However, this time, to prevent repeating the same message, Bob encrypts $m+1$ (i.e., $c' = (m+1)^e \bmod n$). Show how Eve can exploit this protocol to obtain m. (Hint: try to obtain $x = m^2 + m \bmod n$ and then compute $x + 1$ and $x + m^3$).

   (b) An efficient implementation for the decryption of RSA, uses the fact that the recipient knows not only d, but also p and q. The algorithm starts with computing $m_p = c^d \bmod p$ and $m_q = c^d \bmod q$, and then combines the results using the Chinese Reminder Theorem. What is the expected speed-up of this approach with comparison to computing $c^d \bmod n$ directly?

   (c) Consider the case where the recipient decrypted a ciphertext c using the above method, but during decryption $\bmod p$, he obtained a wrong value (i.e., $m'_p \neq c^d \bmod p$), and thus constructed a wrong value m'. Show that an adversary who knows m' and in addition the correct message m, can obtain the factorization of $n = pq$.

   (d) As mentioned in class, choosing $e = 3$ allows for a fast encryption/signature verification which takes at most only two exponentiations. Show that in this case, d has a relatively large value. Explain why the system is still secure.

   (e) It was suggested that for servers which need to decrypt/sign many messages, its better to have lower computational overload, and thus, pick a small d. Explain why picking $d = 3$ is not a good idea.