Introduction to Cryptography – Exercise no. 4

Submit in Pairs/Single to **mailbox** 19 by 13/1/13, 1:00 p.m.

1. In this question, you are requested to compute a Diffie-Hellman key. For this purpose, we shall use $p = 44449$ and $g = 11114$.

   Usually, one first selects the secret keys, and then the public keys are computed. However, we shall start by selecting the public keys, which then will be used to compute the secret keys, and then the joint key.

   Let y_1 be digits 2–5 of the ID of the first submitter, i.e., if your ID is 123456789, $y_1 = 2345$. Let y_2 be digits 6–9 of the ID of the second submitter (unless there is one, and then these are the digits of the ID number of the submitter).

   (a) Find x_1 for which $g^{x_1} \equiv y_1 \bmod p$ and x_2 for which $g^{x_2} \equiv y_2 \bmod p$.

   (b) Compute the Diffie-Hellman key produced by the public keys y_1, y_2.

   (c) Given the public key $y = 12345$, find its corresponding x. Explain how you succeeded to do so, despite the fact that Diffie-Hellman is secure.

2. 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.

3. After the above algorithm was shown to fail, Prof. Ode have found that there are primes for which this algorithm works. Let $p = 2^n + 1$, show that the algorithm works for any generator modulo this sort of prime numbers.
   Hint: show that $g^{10\ldots0_2} \bmod p = -1$, and follow by showing that the two square roots of some quadratic residue a differ only in the most significant bit of their exponents.

4. (a) Calculate the Jacobi symbol $\left(\frac{1253}{78923}\right)$ 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 pair of messages, M_1, M_2 shall be called "switching" pair, if $E_K(M_1) = M_2$ and $E_K(M_2) = M_1$. Prove that in an RSA system with $n = pq$ and a public key (n, e) the number of "switching pairs" is $(gcd(p-1, e^2-1)+1) \cdot (gcd(q-1, e^2-1)+1)$.

5. Consider the problem of computing $gcd(x^{\frac{p-1}{2}}, f(x-\delta))$ (as part of the algorithm for finding roots modulo prime numbers of the form $4k = 1$). Recall that $gcd(p(x), q(x))$ is defined to be a polynomial $r(x)$ of the highest degree, such that $r(x)|p(x)$ and $r(x)|q(x)$ (and is unique up to a linear scale).

   In the lecture we claimed that computing this gcd can be done efficiently, and in this question, we will explore an algorithm for the task.

   (a) Explain why simply dividing $x^{\frac{p-1}{2}}$ by $f(x-\delta)$ (as done in the Euclid algorithm) is not a feasible approach.

   (b) Prove that $gcd(p(x), q(x)) = gcd(p(x) \bmod q(x), q(x))$.

   (c) Show how to compute $x^{\frac{p-1}{2}} \bmod f(x-\delta)$ efficiently. (Hint: consider what changes are needed in the Square and multiply algorithm)

   (d) Explain how to efficiently compute $gcd(x^{\frac{p-1}{2}}, f(x-\delta))$. What is the time complexity of your solution?