Cryptography and Security 2017

Exercise Sheet 5

Exercise 1 Authenticated Diffie-Hellman Key Agreement Protocol

Let us consider a public-key Diffie-Hellman key agreement protocol derived from the simple Diffie-Hellman protocol. In this protocol, we have the following public parameters:

- a large prime p
- a large prime factor q of $p-1$
- an element g of order q in $\mathbf{Z}_p^*$

Each user U has a random secret key $X_U \in \mathbf{Z}_q$ uniformly distributed and a public key $Y_U = g^{X_U} \bmod p$. All the users' public keys are stored in an authenticated database (e.g., using a trusted third party), which is publicly readable. We propose the following key agreement protocol between users A and B.

- A generates $a \in \mathbf{Z}_q$ using a pseudorandom number generator, computes $v = g^a \bmod p$, and sends v to B.
- B generates $b \in \mathbf{Z}_q$ using a pseudorandom number generator, computes $w = g^b \bmod p$ and sends w to A.

In the end, A and B share the secret key $K = g^{aX_B + bX_A} \bmod p$.

1. Explain how A can compute K.
2. Assume the pseudorandom number generator of B is biased in the sense that it only generates small numbers (e.g., of length around 40 bits) instead of generating numbers almost uniformly in $\mathbf{Z}_q$. Show how an adversary A^* can impersonate A to set up a key with B. Suggest a countermeasure.
3. Assume that $b = ac$ for some small c. Show that the adversary A^* can impersonate A and set up a key with B. Suggest a countermeasure.

Exercise 2 Square Roots

1. Let $n := pq$ for primes p, q. Given access to a oracle $\mathcal{O}$ that returns a random square root mod n, explain how to factor n using $\mathcal{O}$.
2. Show that your algorithm returns a correct factorization.
3. How many queries to $\mathcal{O}$ do you need to perform in average? Deduce the expected complexity of your algorithm.

Exercise 3 Modulo 101 Computation

Through *all* this exercise, we will let $p = 101$.

1. Show that p is a prime number.

2. What is the order of $\mathbf{Z}_p^*$?

3. If $x = \sum_{i=0}^{2\ell-1} d_i 10^i$ with $0 \leq d_i < 10$ for all i, show that

$$x \equiv \sum_{i=0}^{\ell-1} (-1)^i (d_{2i} + 10 d_{2i+1}) \pmod{101}$$

   Deduce an algorithm to compute $x \bmod 101$ easily.

4. Show that every element of $\mathbf{Z}_p^*$ has a unique 7th root and give an explicit formula to compute it (recall that $p = 101$).
   Application: Find the 7th root of 2 in $\mathbf{Z}_p^*$.

5. Given $g \in \mathbf{Z}_p^*$ we let $y = g^{10} \bmod p$. Using 3 multiplications modulo p and 2 tests, give an algorithm with input y to decide whether g is a generator or not (recall that $p = 101$).
   Application: show that 2 is a generator.

6. Under which condition is x a quadratic residue in $\mathbf{Z}_p^*$?

7. Show that 5 is a quadratic residue in $\mathbf{Z}_p^*$.

8. Show that 10 is a 4th root of 1 in $\mathbf{Z}_p^*$.

9. Show that for all $y \in \mathbf{Z}_p^*$ we have that $y^{\frac{p-1}{4}}$ is 10^k for some $k \in \{0, 1, 2, 3\}$.

   Show that $y^{\frac{p+3}{4}}$ can be written $y \times 10^k$.

10. Deduce that if x is a quadratic residue then either $x^{\frac{p+3}{8}}$ or $10x^{\frac{p+3}{8}}$ is a square root of x. Provide an algorithm to extract square roots in $\mathbf{Z}_p^*$.

11. Find a square root of 5.