Cryptography 2018 Midterm Solutions

1. Explain the Euclidean algorithm for computing $\gcd(a, b)$ for two integers a and b. Show that if a and b each have at most k bits, then it has running time of $O(k^3)$.

Solution

See notes.

2. Show that if $g^a \equiv_m 1$ and $g^b \equiv_m 1$ then $g^{\gcd(a,b)} \equiv_m 1$.

Solution

We can write $\gcd(a, b)$ as $xa + yb$ for some integers x and y, so

$$g^{\gcd(a,b)} = g^{xa+yb} = (g^a)^x \cdot (g^b)^y \equiv_m 1^x \cdot 1^y = 1.$$

3. Determine whether or not 17 is a generator (primitive root) modulo 23.

Solution

The divisors of 22 are $1, 2, 11$, and 22. So by Fermat's Little Theorem we have to check if either of 17^2 and 17^{11} are 1 mod 23.

- $17^2 \equiv_{23} -6^2 \equiv_{23} 13$
- $17^{11} = -6^{2(4)+3} \equiv_{23} 13^4 \cdot (-6)^3 \equiv_{23} 8^2 \cdot (-6)^3 \equiv_{23} (4 \cdot -6)^3 \equiv (-1)^3 = (-1)$.

So yes, 17 is a generator.

4. For prime p show that $\phi(p^2) = p(p - 1)$.

Solution

As $\phi(n)$ is the number of integers in $1, \ldots, n$ that are relatively prime to n, and the only integers in $1, \ldots p^2$ that are not prime to p^2 are those with a factor of p, it is enough to show that there are only $p^2 - p(p-1) = p$ integers in $1, \ldots, p^2$ that share a factor with p^2. This is clearly $p, 2p, 3p, \ldots, pp$, as the only non-trivial divisor of p^2 is p.

5. Let p be an odd prime and let g be a primitive root modulo p. An integer a has a square root modulo p if there is some integer b such that $b^2 = a$ modulo p. Prove that a has a square root modulo p if and only if $\log_g(a)$ modulo p is even.

Solution

If $\log_g(a)$ is even, so $2e$ for some integer e, then $g^{2e} = a$ and so $b = g^e$ is a square root of a. On the other hand, if $b^2 = a$ modulo p then where $e = \log_g(b)$, (so $1 \leq e \leq p-1$), we have that $g^{2e} = b^2 = a$. Write $2e = c(p-1)+r$ for some r with $1 \leq r \leq p-1$. Then $g^r = g^{2e-c(p-1)} = g^{2e} \cdot g^{(p-1)\cdot c} \equiv g^{2e} \equiv a$. So $\log_g(a) = r$, but $r = 2e - c(p-1)$, so because e and $p-1$ are even, so is r.

6. Recall that in the El-Gamal PKC for a prime p and generator g, there is a public key $A(= g^a)$ where a is secret. A message m is encrypted to $c_1 = g^k$ and $c_2 = mA^k$ for some secret k. (Everything is in $\mathbb{F}_p^*$).

   (a) Knowing the secret key a how would you decrypt (c_1, c_2) to m? (You of course know p, g, and A as well.)

   (b) What is the Diffie-Hellman Problem for a prime p and generator g modulo p.

   (c) Assuming that you have an oracle for the Diffie-Hellman problem, explain how you could crack the El-Gamal PKC: that is, quickly find m without using a.

Solution

(a) Inverting c_1^a modulo p we calculate:

$$c_2(c_1^{-a}) = mA^k g^{-ak} = mg^{ak}g^{-ak} = m.$$

(b) Knowing g and p, we are given $A = g^a$ and $B = g^b$ in $(\mathbb{Z}/p\mathbb{Z})^*$, and must find g^{ab}.

(c) Taking $A = g^a$ and $B = c_1 = g^k$, the oracle for the Diffie-Hellman problem gives us g^{ak}. Inverting this we get g^{-ak} and then multiplying by c_2 we get

$$c_2 g^{-ak} = mA^k g^{-ak} = mg^{ak}g^{-ak} = m,$$

as needed.

7. (a) Use Shank's (giant-steps baby-steps) algorithm to compute $\log_5 15 \mod 23$.

   (b) The element 6 is not a generator mod 23. What can go wrong in Shank's algorithm in computing $\log_6 15 \mod 23$? (One of two things will happen, when will which happen?)

Solution

(a) As $\lceil\sqrt{24}\rceil = 5$ we compute lists:

$L_1 = \{5,\ 5^2 \equiv_{23} 2,\ 5^3 \equiv_{23} 10,\ 5^4 \equiv_{23} 4,\ 5^5 \equiv_{23} 20\}$

$L_2 = \{15,\ 15(-8) \equiv_{23} 18,\ 15(-8)^2 \equiv_{23} 17,\ 15(-8)^3 \equiv_{23} 2 \dots\}$

where we have noticed that $20 \cdot (-8) \equiv_{23} (-3)(-8) = 24 \equiv_{23} 1$ so that $5^{-5} \equiv_{23} -8$. Finding the element $5^2 \equiv_{23} 2 \equiv_{23} 15(-8)^3 = 15/(5^5)^3$ we get that $5^{2+15} = 15$. So in $\mathbb{F}_{23}^*$, $\log_5 15 = 17$.

(b) There can be either no solutions, or many (more than one) solutions to $6^x \equiv_{23} 15$. If there are no solutions there will be no element in both L_1 and L_2. If there are many solutions, the will be several pairs of common elements.

8. Explain what the Polhig Hellman algorithm allows you to do. Explain why it is important to know about this when you are using the Diffe Hellman Key exchange.

Solution

It is an algorithm, using the chinese remainder theorem, that allows one to solve the discrete log problem quickly modulo p if the prime factorisation of $p-1$ has no large primes. The Deffe Hellman key exchange can be solved by computing a discrete log so the Polhig Hellman algorithm tells us we must choose our prime p for the Diffe Hellman key exchange carefully.