COMP0141 Security
Tutorial 3 Solutions

February 2023

Questions

1. Number Theory.

   (a) Find integers x, y such that $gcd(867, 1026) = 867x+1026y$ using the extended Euclidean algorithm.
   Answer: First, we calculate the gcd using the Euclidean algorithm:

$$\begin{aligned} 1026 &= 1 \cdot 867 + 159 \\ 867 &= 5 \cdot 159 + 72 \\ 159 &= 2 \cdot 72 + 15 \\ 72 &= 4 \cdot 15 + 12 \\ 15 &= 1 \cdot 12 + 3 \\ 12 &= 4 \cdot 3 + 0 \end{aligned}$$

The last nonzero remainder is 3, so $gcd(867, 1026) = 3$. Let us now make use of the quotients $q_1 = 1$, $q_2 = 5$, $q_3 = 2$, $q_4 = 4$, $q_5 = 1$, $q_6 = 4$ to determine x and y using the extended Euclidean algorithm. The first step is to form the following sequences:

$$\begin{aligned} &x_0 = 0, x_1 = 1, x_j = -q_{j-1}x_{j-1} + x_{j-2} \\ &y_0 = 1, y_1 = 0, y_j = -q_{j-1}y_{j-1} + y_{j-2} \end{aligned}$$

Then,

$$ax_n + by_n = gcd(a, b)$$

In this example, $n = 6$ and the sequences are as follows:

$$\begin{aligned} &x_0 = 0 && y_0 = 1 \\ &x_1 = 1 && y_1 = 0 \\ &x_2 = -1x_1 + x_0 = -1(1) + 0 = -1 && y_2 = -1y_1 + y_0 = -1(0) + 1 = 1 \\ &x_3 = -5x_2 + x_1 = -5(-1) + 1 = 6 && y_3 = -5y_2 + y_1 = -5(1) + 0 = -5 \\ &x_4 = -2x_3 + x_2 = -2(6) + (-1) = -13 && y_4 = -2y_3 + y_2 = -2(-5) + 1 = 11 \\ &x_5 = -4x_4 + x_3 = -4(-13) + 6 = 58 && y_5 = -4y_4 + y_3 = -4(11) + (-5) = -49 \\ &x_6 = -1x_5 + x_4 = -1(58) + (-13) = -71 && y_6 = -1y_5 + y_4 = -1(-49) + 11 = 60 \end{aligned}$$

Thus,

$$867(-71) + 1026(60) = gcd(867, 1026)$$

An alternative method you may have seen is to work backwards through the remainders from the line revealing the gcd; however, this method only works well for small numbers.

(b) Euler's theorem states that if $gcd(a,n) = 1$, then $a^{\phi(n)} = 1 \pmod{n}$, where ϕ is Euler's totient function. Use it to find the last digit of 7^{2023}.
Answer: This is equivalent to computing $7^{2023} \pmod{10}$.

The elements of $\{1, 2, ..., 10\}$ that are relatively prime to 10 are 1, 3, 7, and 9, so $\phi(10) = 4$.

Now, $7^{2023} = 7^{505 \cdot 4+3} = 7^{505 \cdot \phi(10)+3} = (7^{\phi(10)})^{505} \cdot 7^3 = 1^{505} \cdot 7^3 = 7^3 = 343 = 3 \pmod{10}$, so the last digit is 3.

(c) Find the multiplicative inverse of each of the following: 1, 3, and $5 \pmod{6}$.
Answer: There exists a multiplicative inverse of an integer $a \pmod{n}$ if and only if $gcd(a,n) = 1$. Thus, 3 does not have an inverse $\pmod{6}$.

1 is trivially the inverse of $1 \pmod{6}$.

By Euler's theorem, we have that $5^{\phi(6)} = 1 \pmod{6}$. The elements of $\{1, 2, 3, 4, 5, 6\}$ that are relatively prime to 6 are 1 and 5, so $\phi(6) = 2$. Thus, $5^2 = 1 \pmod{6}$, which means 5 is its own inverse $\pmod{6}$.

You can also compute the inverse using the extended Euclidean algorithm by finding integers x, y such that $5x + 6y = 1$.

Then, $5^{-1} = x \pmod{6}$.

(d) Prove that $n - 1$ is always the multiplicative inverse of $n - 1 \pmod{n}$.
Answer: $(n - 1)(n - 1) = n^2 - 2n + 1 = 1 \pmod{n}$. This is yet another way to show that $5^{-1} = 5 \pmod{6}$ in the last exercise.

(e) Fermat's little theorem states that if p is a prime and p does not divide a, then $a^{p-1} = 1 \pmod{p}$. How does this relate to Euler's theorem?
Answer: $a^{\phi(n)} = 1 \pmod{n} \implies a^{\phi(p)} = 1 \pmod{p} \implies a^{p-1} = 1 \pmod{p}$.

(f) Lagrange's theorem from group theory states that if H is a subgroup of a finite group G, then the order of H divides the order of G. How does this relate to Euler's theorem?
Answer: Consider the multiplicative group of integers modulo n, $G = (\mathbb{Z}/n\mathbb{Z})^\times$, which consists of the elements of $\{1, 2, ..., n\}$ that are relatively prime to n. The number of such elements is $\phi(n)$, which is the order of G. Now, consider the subgroup generated by an element $a \in G$: $\langle a \rangle = \{a^i | i \in \mathbb{Z}\}$. The order of this subgroup is the smallest positive integer k such that $a^k = 1 \pmod{n}$. By Lagrange's theorem, k divides $\phi(n)$; that is, there exists some integer m such that $\phi(n) = m \cdot k$. Then, $a^{\phi(n)} = a^{m \cdot k} = (a^k)^m = 1^m = 1 \pmod{n}$.

(g) Prove that if $gcd(a,n) = 1$, then $x = y \pmod{\phi(n)}$ implies $a^x = a^y \pmod{n}$.
Answer: $x = y \pmod{\phi(n)} \implies x = y + k \cdot \phi(n)$ for some integer k.
Then, $a^x = a^{y+k \cdot \phi(n)} = a^y (a^{\phi(n)})^k = a^y(1^k) = a^y \pmod{n}$.

2. Cryptographic Hardness Problems.

The security of cryptographic systems is often based on the intractability of certain "hard" problems. Let g be a generator of a cyclic group of order q, and let x, y be random elements of $\mathbb{Z}/q\mathbb{Z}$.

- Discrete logarithm problem (DL): Given g and g^x, find x.
- Computational Diffie-Hellman problem (CDH): Given g, g^x, and g^y, compute g^{xy}.
- Decisional Diffie-Hellman problem (DDH): Given g, g^x, g^y, g^z, decide if $g^z = g^{xy}$ or g^r for some random $r \in \mathbb{Z}/q\mathbb{Z}$.

These problems are related through (polynomial-time) reductions, meaning that an algorithm that solves one problem (in polynomial time) can be used by an algorithm to solve another problem (in polynomial time). Show that if you can solve the DL problem, then you can solve the CDH problem. Show that if you can solve the CDH problem, then you can solve the DDH problem.
Answer: DL → CDH: Given g, g^x, and g^y, solve the DL problem to obtain x. Then, compute $(g^y)^x = g^{xy}$ as desired.

CDH → DDH: Given g, g^x, g^y, g^z, solve the CDH problem to obtain g^{xy}. Then, check if $g^z = g^{xy}$

3. RSA Cryptosystem.

   The RSA cryptosystem works as follows:

   - Alice chooses secret primes p and q and computes $n = pq$.
   - Alice chooses an encryption exponent e such that $gcd(e, (p-1)(q-1)) = 1$.
   - Alice computes the decryption exponent d such that $de = 1 \pmod{(p-1)(q-1)}$.
   - Alice makes (n, e) public, but keeps (p, q, d) secret.
   - Bob encrypts his message m as $c = m^e \pmod{n}$ and sends c to Alice.
   - Alice decrypts the ciphertext c by computing $m = c^d \pmod{n}$.

   Consider a toy example in which the public values are $n = 91$ and $e = 35$.

   (a) Encrypt the message $m = 15$.
   Answer: $m^e = 15^{35} \pmod{91}$ We can calculate this using repeated squaring:

   $$\begin{aligned} 15^2 &= 43 \pmod{91} \\ (15^3 &= 15^2 * 15 = 43 * 15 = 8 \pmod{91}) \\ 15^4 &= 43^2 = 29 \pmod{91} \\ 15^8 &= 29^2 = 22 \pmod{91} \\ 15^{16} &= 22^2 = 29 \pmod{91} \\ 15^{32} &= 29^2 = 22 \pmod{91} \end{aligned}$$

   Then, $15^{35} = 15^{32+3} = (15^{32})(15^3) = (22)(8) = 85 \pmod{91}$. You can also use the Chinese remainder theorem to solve this problem.

   (b) Suppose an attacker knows that a ciphertext c encrypts either a message m_1 or a message m_2. How can they determine which one?
   Answer: The attacker can compute m_1^e and m_2^e and check which one equals c.

   (c) Parameters for RSA must be chosen carefully. In this toy example, the decryption exponent d can be recovered easily. Find d.
   Answer: We can factor $n = 91$ into $p = 7$ and $q = 13$. Now, $\phi(91) = \phi(7)\phi(13) = 6 \cdot 12 = 72$. Since $de = 1 \pmod{(p-1)(q-1)}$, we have that $d(35) + k(72) = 1$, and we can find d using the extended Euclidean algorithm:

   $$\begin{aligned} 72 &= 2 \cdot 35 + 2 \\ 35 &= 17 \cdot 2 + 1 \\ 2 &= 2 \cdot 1 + 0 \end{aligned}$$

   So,

   $$\begin{aligned} x_0 &= 0 \\ x_1 &= 1 \\ x_2 &= -2(1) + 0 = -2 \\ x_3 &= -17(-2) + 1 = 35 \end{aligned}$$

   Therefore, $d = 35$.

4. ElGamal Encryption.

   ElGamal encryption works as follows:

   - Alice chooses a cyclic group G of order q and a generator g.

- Alice chooses a random $x \in \{1, ..., q-1\}$ and computes $h = g^x$.
- Alice makes (G, q, g, h) public, but keeps x secret.
- Bob chooses a random $y \in \{1, ..., q-1\}$ and computes g^y and h^y.
- Bob encrypts his message m as $(c_1, c_2) = (g^y, m \cdot h^y)$ and sends (c_1, c_2) to Alice.
- Alice decrypts the ciphertext (c_1, c_2) by computing $c_2 \cdot c_1^{-x}$.

(a) Show that ElGamal encryption is correct.
Answer: $c_2 \cdot c_1^{-x} = (m \cdot h^y) \cdot (g^y)^{-x} = (m \cdot (g^x)^y) \cdot g^{-xy} = m$.

(b) Left-or-right IND-CPA security means an adversary who can pick two arbitrary plaintexts m_1, m_2 should not be able to distinguish the encryption of one of them from the encryption of the other. Real-or-random IND-CPA security means an adversary who can pick an arbitrary plaintext m should not be able to distinguish the encryption of it from the encryption of a random message. (Both definitions are proven to be equivalent.) Show that if the DDH problem is hard, then ElGamal encryption is (real-or-random) IND-CPA secure.
Answer: The elements g^x and g^y are chosen at random. It is difficult to distinguish $h^y = g^{xy}$ from random (DDH). Thus, it is difficult to distinguish $m \cdot h^y$ from random.

(For more detail about the equivalence between Left-or-right IND-CPA and Real-or-random IND-CPA please refer to `https://web.cs.ucdavis.edu/~rogaway/papers/sym-enc.pdf`.)

(c) IND-CCA security means an adversary who can pick two arbitrary plaintexts m_1, m_2 should not be able to distinguish the encryption of one of them from the encryption of the other, even when given the additional ability to see decryptions of chosen ciphertexts (other than those of m_1 and m_2). Show that ElGamal encryption is not IND-CCA secure.
Answer: Given a cipertext (c_1, c_2), an adversary can query the decryption of a related ciphertext $(c_1, m' \cdot c_2)$ for a message m' he knows. The decryption will yield $m' \cdot m$, from which he can compute m and compare it to m_1 and m_2.

(d) Why doesn't the attack from Problem 3b work on ElGamal encryption?
Answer: An attacker does not know the random values y_1, y_2 used to encrypt m_1 and m_2 and thus cannot compare their encryptions to the given ciphertext. Also, this is precisely a form of chosen plaintext attack, against which ElGamal is secure by part b.