15-251: Great Theoretical Ideas In Computer Science

Recitation 11 Solutions

Announcements

- We have a midterm next Wednesday (November 18).

Number theory warm-up

(a) Why is $\mathbb{Z}_n^*$ closed under multiplication? (Prove that $x, y \in \mathbb{Z}_n^* \implies xy \in \mathbb{Z}_n^*$.)

(b) Prove that for any two distinct primes p, q, we have $\phi(pq) = (p-1)(q-1)$. (What about $\phi(p^2)$?)

(a) By definition of $\mathbb{Z}_n^*$, we have that $\gcd(a, n) = \gcd(b, n) = 1$, meaning that neither a nor b has any prime factors in common with n. The set of prime factors of ab is just the union of the prime factors of a and the prime factors of b, so ab will also share no prime factors with n.

(b) Recall that $\phi(pq)$ is the number of natural numbers strictly less than pq that are relatively prime to pq. For any number i in the range $[0, pq-1]$, we know that $\gcd(i, p)$ is 1 iff i is not an integer multiple of p. Among the pq numbers in this range, there are q multiples of p, p multiples of q, and one number that is a multiple of both. Because p, q are prime, anything else is relatively prime to pq, so $\phi(pq) = pq - p - q + 1 = (p-1)(q-1)$.
The p^2 case is even easier: everything that's not an integer multiple of p is relatively prime to p^2, so $\phi(p^2) = p^2 - p$.

Actual interesting stuff (part 1)

Understand Diffie-Hellman key exchange and be able to answer the questions below.

Alice		Bob
Picks a prime p	(1)	
Picks a generator $B \in \mathbb{Z}_P^*$	(2)	
Randomly draws $E_1 \in \mathbb{Z}_{\phi(P)}$	(3)	
Computes $B^{E_1} \in \mathbb{Z}_P^*$	(4)	
Sends $P, B \in \mathbb{Z}_P^*, B^{E_1} \in \mathbb{Z}_P^*$	(5)	Receives $P, B \in \mathbb{Z}_P^*, B^{E_1} \in \mathbb{Z}_P^*$
	(6)	Randomly draws $E_2 \in \mathbb{Z}_{\phi(P)}$
	(7)	Computes $B^{E_2} \in \mathbb{Z}_P^*$
	(8)	Sends $P, B \in \mathbb{Z}_P^*, B^{E_2} \in \mathbb{Z}_P^*$
Compute $(B^{E_2})^{E_1} = B^{E_1 E_2} \in \mathbb{Z}_P^*$	(9)	Compute $(B^{E_1})^{E_2} = B^{E_1 E_2}$

- In lines 3 and 5, where we pick random exponents, why are we picking from $\mathbb{Z}_{\phi(P)}$ rather than $\mathbb{Z}_P^*$ or $\mathbb{Z}$?

- Lines 4, 6, and 9 involve modular exponentiation. How can we accomplish this efficiently?
- What information do we assume an eavedropper wouldn't be able to derive? Why would it be bad if an eavesdropper could derive this information?
- In line 2, why do we need a generator? Why not any element?
- I thought cryptography was about sending messages. Why are Alice and Bob trying to secretly agree on an element of $\mathbb{Z}_P^*$?

- Remember Euler's theorem! When we are exponentiating in $\mathbb{Z}_P^*$, the exponent effectively lives in $\mathbb{Z}_{\phi(P)}$.
- Use the repeated squaring method.
- Since the eavesdropper might learn B^{E_1}, B^{E_2} when these values are sent over, ideally they won't be able to figure out E_1 or E_2. Luckily, the only way to do so seems to be to compute a discrete logarithm, which is thought to be computationally hard.
- If B is not a generator, then the discrete logarithm might be easy to compute. (Conceptual example: $B = 3$ in $\mathbb{Z}_4$.)
- The key, which now is known only to Alice and Bob, can be used to encrypt messages via a one-time pad protocol.

Actual interesting stuff (part 2)

Understand RSA and be able to answer the questions below.

Alice		Bob
	(1)	Secretly chooses large primes P, Q
	(2)	Computes $N = PQ$ and chooses $E \in \mathbb{Z}_{\phi(N)}^*$
	(3)	Secretly computes $D = E^{-1} \in \mathbb{Z}_{\phi(N)}^*$
Prepares to encrypt message M	(4)	Publishes (N, E)
Computes $C = M^E \in \mathbb{Z}_N^*$	(5)	
Sends C	(6)	Receives C
	(7)	Computes $C^D = (M^E)^D = M^{ED} = M^{EE^{-1}} = M$

- Why are E, D chosen from $\mathbb{Z}_{\phi(N)}^*$ rather than $\mathbb{Z}_{\phi(N)}$?
- How does Bob efficiently invert E in line 3?
- Why doesn't Bob just pick a random big number N?
- What would happen if Bob published P, Q too?
- In what ways could an eavesdropper attempt to crack the encryption? Why is it hard to do so?

- Remember Euler's theorem! When we are exponentiating in $\mathbb{Z}_N^*$, exponents effectively live in $\mathbb{Z}_{\phi(N)}$ Also, E needs to have a multiplicative inverse, which is only a property of elements of $\mathbb{Z}_{\phi(N)}^*$.
- The extended Euclidean algorithm is an efficient way to find modular multiplicative inverses.
- Bob needs to calculate $\phi(N)$. We don't know how to do this without knowing the factors of N.
- If a malicious party knew P, Q, then they could easily calculate $\phi(N)$ and then re-derive Bob's private key D. (How?)
- If an eavesdropper knew E and "overheard" C, they could try to take the modular Eth root of C. There's no known way to do this efficiently, though. If an eavesdropper efficiently computed $\phi(N)$, they could derive Bob's private key, but in general ϕ is hard to compute. (It's easier if you know the factorization of the number, but factoring is also thought to be computationally hard.)