15-251: Great Theoretical Ideas In Computer Science

Recitation 14

Diffie Hellman

Recall the Diffie-Hellman protocol for securely generating a secret key over a public communication channel:

Apoorva		Bhagwat
Picks a large 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, B^{E_1}	(5)	Receives P, B, B^{E_1}
	(6)	Randomly draws $E_2 \in \mathbb{Z}_{\phi(P)}$
	(7)	Computes $B^{E_2} \in \mathbb{Z}_P^*$
Receives B^{E_2}	(8)	Sends B^{E_2}
Computes $(B^{E_2})^{E_1} = B^{E_1 E_2} \in \mathbb{Z}_P^*$	(9)	Computes $(B^{E_1})^{E_2} = B^{E_1 E_2} \in \mathbb{Z}_P^*$

- In line 2, why must B be a generator?
- In lines 3 and 5, why are the random exponents chosen from the set $\mathbb{Z}_{\phi(P)}$?
- Lines 4, 6, and 9 involve modular exponentiation. How can we accomplish this efficiently?
- An eavesdropper can obtain $B, B^{E_1}, B^{E_2} \in \mathbb{Z}_P^*$. Can she efficiently recover $B^{E_1 E_2}$?
- Why is this protocol useful?

ElGamal

The ElGamal encryption system is a way of using the Diffie-Hellman protcol to exchange encrypted messages. Suppose Apoorva wants to send a message M to Bhagwat.

Apoorva		Bhagwat
	(1)	Picks a large prime P
	(2)	Picks a generator $B \in \mathbb{Z}_P^*$
	(3)	Randomly draws $E_1 \in \mathbb{Z}_{\phi(P)}$
	(4)	Computes $B^{E_1} \in \mathbb{Z}_P^*$
Receives P, B, B^{E_1}	(5)	Sends P, B, B^{E_1}
Randomly draws $E_2 \in \mathbb{Z}_{\phi(P)}$	(6)	
Encode M as an element of $\mathbb{Z}_P^*$	(7)	
Computes $B^{E_2}, MB^{E_1 E_2} \in \mathbb{Z}_P^*$	(8)	
Sends $(B^{E_2}, MB^{E_1 E_2})$	(9)	Receives $(B^{E_2}, MB^{E_1 E_2})$
	(10)	Computes $(B^{E_2})^{E_1} = B^{E_1 E_2} \in \mathbb{Z}_P^*$
	(11)	Computes $(B^{E_1 E_2})^{-1} \in \mathbb{Z}_P^*$
	(12)	Computes $(MB^{E_1 E_2})(B^{E_1 E_2})^{-1} = M \in \mathbb{Z}_P^*$

Suppose $P = 17$, $B = 3$. Bhagwat sends Apoorva $(17, 3, 6)$ (line 5) (Note: $6 = 3^{15}$). Apoorva sends back $(7, 1)$ (line 9). What is the decrypted message?

RSA

Receiver Protocol

1. Choose two large *distinct* primes P and Q
2. Compute $N = PQ$ and $\phi(N) = (P-1)(Q-1)$
3. Choose $E \in \mathbb{Z}_{\phi(N)}^*$
4. Publish the *public key*: (N, E)
5. Compute the decyption key $D = E^{-1} \in \mathbb{Z}_{\phi(N)}^*$
6. Upon receipt of ciphertext C, compute $M = C^D \in \mathbb{Z}_N^*$

Sender Protocol

1. Encode M as an element of $\mathbb{Z}_N^*$
2. Send $M^E \in \mathbb{Z}_N^*$

- Why must P and Q be distinct?
- Why must the encrpytion key E be an element of $\mathbb{Z}_{\phi(N)}^*$?
- How does the receiver compute the decprytion key D?
- Given ciphertext C, why is C^D equal to the original message?
- What if $M \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$? Is this something the receiver needs to worry about?