ECC

Day 1: Went over arithmetic
Day 2: Went over code
Day 3: How to apply to cryptography

Analog to Diffie Hellman Key Exchange

Given $g^a, g$ and $p$
a is hard to find

Alice → $g^a$ → Bob($g^b$)$^b$
Bob → $g^b$ → Alice($g^a$)$^a$

Public elements $\mathbb{F}_p^*$
Curve $E_p(a,b)$
Point $G$ on Curve
Order of $G$ is large.
Let $n$ be order of $G$.

Alice chooses $n_A < n$
Sends Bob $n_A \cdot G$.
Bob chooses $n_B < n$
Sends Alice $n_B \cdot G$.

Alice take $n_A \cdot (n_B \cdot G) = (n_A \cdot n_B) \cdot G$
Bob take $n_B \cdot (n_A \cdot G) = (n_A \cdot n_B) \cdot G$

Shared Key
Public Key Cryptography Elliptic Curves
Similar to El Gamal

Alice picks a curve $E_{\mathbb{F}}(a,b)$ Public keys

She picks a secret value $n_A$ Private key

She calculates point $P_A$ Public key

Chooses a point $G$ with a high order $n$

Post Public key $P_A = n_A \times G$ Public

Bob sends $MSG \Rightarrow Alice$

1. Bob's Plaintext is $P_m$ (a pt on curve)
2. Picks secret $k < n$
3. $C_1 = k \times P_A \times G = (k \times n_A) \times G$
4. $C_2 = P_m + k \times P_A = (P_m + (k \times n_A) \times G$)

When Alice receives this, she does
1. temp = $n_A \times C_1 = (n_A \times k) \times G$
2. $P_m = C_2 - \text{temp}$

$P_m + k \times P_A - n_A \times C_1$

$P_m + k \times n_A \times G - n_A \times k \times G$

$P_m$