Highlights of Chapter 6.

Chapter 6: Elliptic Curves and Cryptography

1. **Number theory concepts:**

   (a) An *elliptic curve* over a field F (where $F \neq \mathbb{F}_2$) is a curve given by a Weierstrass equation $y^2 = x^3 + Ax + B$, with $A, B \in F$, such that $4A^3 + 27B^2 \neq 0$ in F.

   (b) The geometric secant and tangent method on an elliptic curve E to find a third point R on E from two known points P and Q. Addition of points on the elliptic curve.

   (c) Elliptic curve addition algorithm: let E be given by $y^2 = x^3 + Ax + B$, and let P and Q be points on E.

   - If $P = \mathcal{O}$, then $P \oplus Q = Q$. If $Q = \mathcal{O}$, then $P \oplus Q = P$.
   - If $P = (x_1, y_1)$ and $Q = (x_1, -y_1)$, then $P \oplus Q = \mathcal{O}$, i.e., $Q = -P$.
   - If $P \neq Q$ and $P = (x_1, y_1)$, $Q = (x_2, y_2)$, then define $\lambda = (y_2 - y_1)/(x_2 - x_1)$. Then $P \oplus Q = (x_3, y_3)$, with

   $$x_3 = \lambda^2 - x_1 - x_2 \quad \text{and} \quad y_3 = \lambda(x_1 - x_3) - y_1.$$

   - If $P = Q = (x_1, y_1)$, then define $\lambda = (3x_1^2 + A)/(2y_1)$. Then $2P = (x_3, y_3)$ where the coordinates x_3, y_3 are defined as above.

   (d) If E is an elliptic curve over $\mathbb{F}_p$, with p prime, then

   $$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p^2 : y^2 \equiv x^3 + Ax + B \bmod p\} \cup \{\mathcal{O}\}.$$

   (e) Hasse's theorem: if E is an elliptic curve over $\mathbb{F}_p$, then

   $$p + 1 - 2\sqrt{p} \leq \#E(\mathbb{F}_p) \leq p + 1 + 2\sqrt{p}.$$

2. **Cryptography:**

   (a) The Elliptic Curve Discrete Logarithm Problem (ECDLP): given an elliptic curve E over $\mathbb{F}_p$ and points P and Q on E, find a number n such that $nP = Q$, where $nP = P \oplus \cdots \oplus P$ is the sum of n copies of P using the elliptic curve addition algorithm.

   (b) The double-and-add algorithm to compute a multiple of a point on an elliptic curve.

   (c) Collision algorithm to find a solution to an ECDLP problem: to solve $Q = nP$, find lists:

   - List 1: $k_1P, k_2P, \ldots, k_rP$, where $k_1, \ldots, k_r$ are distinct integers.
   - List 2: $k_1'P + Q, k_2'P + Q, \ldots, k_r'P + Q$, where $k_1', \ldots, k_r'$ are distinct integers.

   If $k_uP = k_v'P + Q$, then $Q = (k_u - k_v')P$. One needs about $r \approx 3\sqrt{p}$ to have a "very good chance" of finding a collision.

   (d) Elliptic Diffie-Hellman Key Exchange:

   - A trusted party chooses a large prime p, an elliptic curve E over $\mathbb{F}_p$, and a points P in $E(\mathbb{F}_p)$.
   - Alice chooses a secret integer n_A, Bob chooses a secret integer n_B.
   - Alice computes $Q_A = n_A \cdot P$ and sends it to Bob. Bob computes $Q_B = n_B \cdot P$ and sends it to Alice.
   - Alice computes the secret shared point $n_A \cdot Q_B$. Bob computes the secret shared point $n_B \cdot Q_A$. We have $n_An_BP = n_AQ_B = n_BQ_A$.

   (e) Elliptic Elgamal cryptosystem:

   - A trusted party chooses a large prime p, an elliptic curve E over $\mathbb{F}_p$, and a points P in $E(\mathbb{F}_p)$.

- Alice chooses a private key n_A. Computes $Q_A = n_A \cdot P$ in $E(\mathbb{F}_p)$. Publishes Q_A.
- Bob chooses plaintext $M \in E(\mathbb{F}_p)$, chooses a random element k, and computes $C_1 = kP$ and $C_2 = M + kQ_A$. Sends the ciphertext (C_1, C_2) to Alice.
- Alice computes the plaintext $M = C_2 - n_A C_1 \in E(\mathbb{F}_p)$.

(f) Elliptic Curve Digital Signatures:

- A trusted party chooses a large prime p, an elliptic curve E over $\mathbb{F}_p$, and a points G in $E(\mathbb{F}_p)$ of large prime order q.
- Sam chooses a secret signing key $1 < s < q - 1$. Computes $V = sG$ in $E(\mathbb{F}_p)$, and publishes V.
- Sam chooses a document $d \bmod q$, chooses a random element $e \bmod q$, computes eG in $E(\mathbb{F}_p)$, and a signature $(s_1, s_2) = (x(eG) \bmod q, (d + s \cdot s_1)e^{-1} \bmod q)$. Publish $(d, (s_1, s_2))$.
- Victor computes $v_1 \equiv ds_2^{-1} \bmod q$ and $v_2 \equiv s_1 s_2^{-1} \bmod q$. Then verifies that

$$x(v_1 G + v_2 V) \bmod q = s_1.$$

(g) Lenstra's Factorization Algorithm:

i. Input: N to be factored.
ii. Choose random A, a, and $b \bmod N$.
iii. Set $P = (a, b)$ and $B \equiv b^2 - a^3 - Aa \bmod N$, and $E : y^2 = x^3 + Ax + B \bmod N$.
iv. Loop $j = 2, 3, 4, \ldots$
    A. Set $Q \equiv j \cdot P \bmod N$ and set $P = Q$.
    B. If computing $j \cdot P$ in Step 4 fails, we have found a divisor $d > 1$ of N.
        - If $d < N$, then success, return d.
        - If $d = 1$, then go to step 1.
    C. If computing $j \cdot P$ is successful, then increase j by 1, and return to Step A.