Cryptography Cheat Sheet

RSA Cryptosystem

Key Generation:

1. Select two large primes p and q.
2. Compute $n = p \times q$.
3. Compute Euler's totient function: $\phi(n) = (p-1) \times (q-1)$.
4. Choose public exponent e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$.
5. Compute private exponent d such that $d \equiv e^{-1} \mod \phi(n)$.

Encryption:

$$C = M^e \mod n$$

Decryption:

$$M = C^d \mod n$$

Correctness:

$$M = (M^e)^d \mod n = M \mod n, \text{ because } (e \times d) \equiv 1 \mod \phi(n).$$

ElGamal Cryptosystem

Key Generation:

1. Choose a prime p and a generator g of Z_p^*.
2. Choose a secret key x, where $1 \leq x \leq p-2$.
3. Compute public key $y = g^x \mod p$.

Encryption:

1. Choose a random k such that $1 \leq k \leq p-2$.
2. Compute $c_1 = g^k \mod p$, $c_2 = M \times y^k \mod p$.
3. Ciphertext is (c_1, c_2).

Decryption:

$$M = c_2 \times (c_1^x)^{-1} \mod p$$

Correctness:

$$M = (M \times (y^k)^x)^{-1} \mod p = M \mod p, \text{ because } (g^x)^k = y^k.$$

Elliptic Curve Cryptosystem

Key Generation:

1. Choose a prime p and an elliptic curve $y^2 = x^3 + ax + b$ over F_p.
2. Choose a base point G with order n.
3. Choose a private key d where $1 \leq d < n$.
4. Compute public key $Q = dG$.

Encryption:

1. Choose a random k.
2. Compute $C_1 = kG$, $C_2 = M + kQ$.
3. Ciphertext is (C_1, C_2).

Decryption:

$$M = C_2 - d \times C_1$$

Correctness:

$$M = (M + kQ) - d \times kG = M \mod p, \text{ because } dG = Q.$$

RSA Signature Scheme

Key Generation:

1. Select two large primes p and q.
2. Compute $n = p \times q$.
3. Compute Euler's totient function: $\phi(n) = (p-1) \times (q-1)$.
4. Choose public exponent e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$.
5. Compute private exponent d such that $d \equiv e^{-1} \mod \phi(n)$.

Signing:

$$S = M^d \mod n$$

Verification:

$$S^e \mod n = M$$

Correctness:

$$(M^d)^e \mod n = M \mod n, \text{ because } (e \times d) \equiv 1 \mod \phi(n).$$

ElGamal Signature Scheme

Key Generation:

1. Choose a prime p and a generator g of Z_p^*.
2. Choose a secret key x, where $1 \leq x \leq p-2$.
3. Compute public key $y = g^x \mod p$.

Signing:

1. Choose a random k such that $1 \leq k \leq p-2$ and $\gcd(k, p-1) = 1$.
2. Compute $r = g^k \mod p$, $s = k^{-1} \times (H(M) - xr) \mod (p-1)$.
3. Signature is (r, s).

Verification:

1. Compute $v_1 = (y^r \times r^s) \mod p$.
2. Compute $v_2 = g^{H(M)} \mod p$.
3. If $v_1 = v_2$, the signature is valid.

Correctness:

$$r = g^k \mod p, \quad s = k^{-1} \times (H(M) - xr) \quad \text{so} \quad (g^k)^x \mod p = H(M).$$

Schnorr Signature Scheme

Key Generation:

1. Choose a prime p and a generator g of Z_p^*.
2. Choose a secret key x, where $1 \leq x \leq p-2$.
3. Compute public key $y = g^x \mod p$.

Signing:

1. Choose a random k, where $1 \leq k \leq p-2$.
2. Compute $r = g^k \mod p$, $e = H(r, M) \mod p$.
3. Compute $s = k - ex \mod (p-1)$.
4. Signature is (r, s).

Verification:

1. Compute $v_1 = g^s \times y^r \mod p$.
2. Compute $v_2 = g^e \mod p$.
3. If $v_1 = v_2$, the signature is valid.

Correctness:

$(g^s \times y^r \mod p) = g^e \mod p$ because of the relation with the secret key x.

DSS (Digital Signature Standard)

Key Generation:

1. Choose a prime p and a prime q such that q divides $p-1$.
2. Choose a generator g of order q in Z_p^*.
3. Choose a secret key x, where $0 \leq x < q$.
4. Compute public key $y = g^x \mod p$.

Signing:

1. Choose a random k such that $0 < k < q$.

2. Compute $r = (g^k \mod p) \mod q$.

3. Compute $s = k^{-1} \times (H(M) + xr) \mod q$.

4. Signature is (r, s).

Verification:

1. Compute $v_1 = (y^r \times r^s) \mod p$.

2. Compute $v_2 = g^{H(M)} \mod p$.

3. If $v_1 = v_2$, the signature is valid.

Correctness:

$$(g^k \mod p)^x = H(M) \mod p \text{ through correct relation with } r, s.$$