Cryptography Theory Sheet with Equations

1️⃣ RSA Product Cipher (Public-Key Encryption)

Purpose: Securely send messages; only the receiver can decrypt.

Steps & Equations:

1. **Select two primes:**

$$p, q$$

2. **Compute modulus:**

$$n = p \cdot q$$

3. **Compute Euler's totient:**

$$\phi(n) = (p - 1) \cdot (q - 1)$$

4. **Choose public exponent e:**
   - Must satisfy:

$$\gcd(e, \phi(n)) = 1$$

5. **Compute private key d:**
   - Solve modular inverse:

$$e \cdot d \equiv 1 \bmod \phi(n)$$

6. **Encryption:**

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

7. **Decryption:**

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

Reasoning:

- $d$ensures $M^{ed} \equiv M \bmod n$.
- Modular exponentiation avoids large number overflow.

---

2 RSA Known Plaintext Attack

Purpose: Show vulnerability if plaintext-ciphertext pair is known.

Equations:

1. Known pair:

$$C_1 = P_1^e \bmod n$$

2. Compute target ciphertext:

$$C_2 = P_2^e \bmod n$$

Reasoning:

- If attacker knows (P1, C1), they can analyze patterns.
- Highlights the importance of padding and randomness in RSA.

---

3 RSA Digital Signature

Purpose: Authenticate message and prove integrity.

Equations:

1. Signature generation:

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

2. Signature verification:

$$V = S^e \bmod n$$

3. Check:

$$V = ?\, M$$

Reasoning:

- Only private key holder can generate valid signature.
- Public key allows verification without revealing private key.

---

4 ElGamal Digital Signature

Purpose: Secure message signing using discrete logarithms.

Steps & Equations:

1. Choose prime p and generator g.
2. Private key: x, Public key:

$$y = g^x \bmod p$$

3. Choose random $k$such that $\gcd(k, p-1) = 1$.
4. Compute:

$$r = g^k \bmod p$$
$$s = k^{-1} \cdot (M - x \cdot r) \bmod (p-1)$$

5. Verification:

$$g^M \bmod p = ?\, y^r \cdot r^s \bmod p$$

Reasoning:

- Random k ensures **unique signature**.
- Modular inverse ensures the math holds.
- Hardness of discrete logarithm guarantees security.

---

5 ElGamal Product Cipher (Encryption)

Purpose: Encrypt messages probabilistically.

Equations:

1. Public key: (p, g, y), Private key: x

$$y = g^x \bmod p$$

2. Choose random k.

3. Compute ciphertext:

$$C_1 = g^k \bmod p$$
$$C_2 = M \cdot y^k \bmod p$$

4. Decrypt:

$$M = C_2 \cdot C_1^{p-1-x} \bmod p$$

Reasoning:

- Random $k$makes encryption **probabilistic**, preventing pattern attacks.

---

6 ElGamal Re-randomization (Cipher Refresh)

Purpose: Change ciphertext without changing underlying message.

Equations:

1. Original ciphertext: (C_1, C_2)

2. Re-randomize with new k_2:

$$R_1 = C_1 \cdot g^{k_2} \bmod p$$
$$R_2 = C_2 \cdot y^{k_2} \bmod p$$

Reasoning:

- Message remains same.

- Prevents linking repeated messages → **forward secrecy**.

7️⃣ Vernam Cipher (XOR Symmetric Key)

Purpose: Encrypt/decrypt using XOR.

Equations:

1. Encryption:

$$C_i = P_i \oplus K_i$$

2. Decryption:

$$P_i = C_i \oplus K_i$$

Reasoning:

- Symmetric key encryption.
- Perfect secrecy if key is random and same length as message.

8️⃣ Caesar Cipher (Substitution Cipher)

Purpose: Shift letters for encryption.

Equations:

1. Encryption:

$$C_i = (P_i + \text{shift}) \bmod 26$$

2. Decryption:

$$P_i = (C_i - \text{shift} + 26) \bmod 26$$

Reasoning:

- Simple substitution cipher.
- Demonstrates basic principles of encryption.

Extra Notes on Functions & Values

Function	Purpose
modExp(base, exp, mod)	Efficiently compute $base^{exp} \bmod mod$ without overflow
modInverse(a, m)	Find inverse such that $a \cdot a^{-1} \equiv 1 \bmod m$; essential for private keys and signatures
Random k in ElGamal	Ensures signatures and ciphertexts are unique; prevents attacks on repeated messages
Prime numbers p, q	Security basis for RSA and ElGamal; factoring large numbers is hard
Public & private keys	Public key: for encryption/verification; Private key: for decryption/signing