Symmetric Encryption
$Gen(\lambda) \to k$
$Enc(\lambda, k, m) \to c$
$Dec(\lambda, k, c) \to m'$

Perfect Indistinguishability (PI)
$P[Enc(k, m_0) = c] = P[Enc(k, m_1) = c]$
$\left|P\left[A\left(Enc(k, m_0)\right) = 1\right] - P\left[A\left(Enc(k, m_1)\right) = 1\right]\right| = 0$

- Limitation
  - Satisfiability: $|K| \geq |M|$

Computational Indistinguishability (CI)
$\left|P\left[A\left(Enc(k, m_0)\right) = 1\right] - P\left[A\left(Enc(k, m_1)\right) = 1\right]\right| < \epsilon$

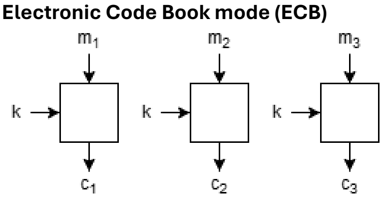
Pseudo-Random Generator (PRG)
- Pseudo-Randomness
  - Indistinguishable from uniform distribution
- Deterministic: $G: \{0,1\}^{\lambda} \to \{0,1\}^{m(\lambda)}$
- Expansion: $m(\lambda) > \lambda$ (The larger the better)

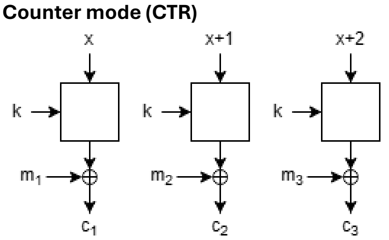
Stream Ciphers
$Gen(\lambda) \to k$
$Enc(\lambda, k, m) \to G(k) \oplus m = c$
$Dec(\lambda, k, c) \to G(k) \oplus c = m'$
- Confusion: Each bit of c depends on multiple bits of k
- Lack of structure: prevent algorithms from exploiting them to break the construction
- Issues: Message Length
  - Solution: Extending PRGs
  - $G'(s_1, \ldots, s_n) = G(s_1) \parallel \cdots \parallel G(s_n)$
  - $(b_i, s_{i+1}) \leftarrow G(s_i)$
- Issues: Key Reuse
  - $c_0 \oplus c_1 = m_0 \oplus m_1$
- Issues: Integrity
  - $(G(k) \oplus m) \oplus \Delta = G(k) \oplus (m \oplus \Delta)$
- CI-Secure
- Not CPA-Secure

Pseudo-Random Functions (PRF)
$F_k = \{f_k: \{0,1\}^{\lambda} \to \{0,1\}^{\lambda}\}$
$Gen(\lambda) \to k = f_k$
$Eval(\lambda, k, x) \to y = f_k(x)$
- CPA-Secure

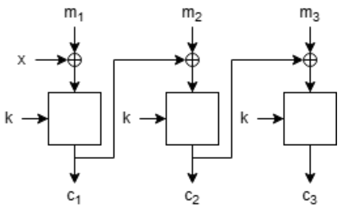
Pseudo-Random Permutations (PRP)
Similar to PRF, except with:
$Invert(\lambda, k, y) \to x' = f_k^{-1}(y)$
- CI-Secure
- Not CPA-Secure

Block Ciphers (uses PRP)
$Enc(\lambda, k, m) \to c = f_k(m)$
$Dec(\lambda, k, c) \to m' = Invert(\lambda, k, c) = f_k^{-1}(c)$
- CI-Secure
- Not CPA-Secure

Electronic Code Book mode (ECB)


Counter mode (CTR)


- CPA-Secure
- Not parallelisable

Cipher Block Chaining mode (CBC)


- CPA-Secure
- Not parallelisable

Message-Authentication Code (MAC)
$Gen(\lambda) \to k$
$Mac(\lambda, k, m) \to t$
$Verify(\lambda, k, m, t) \to$ accept or reject

Universal Hash Functions (UHF)
$H_n = \{h_k: \{0,1\}^n \to \{0,1\}^{\lambda}\}$
$Gen(\lambda) \to k = h_k$
$Eval(\lambda, k, x) \to y = h_k(x)$

Hash Functions
Same as UHF, except with:
- Compression: $n > \lambda$

Collision Resistance (x, x')
For random k, hard to find $x \neq x'$ s.t. $h_k(x) = h_k(x')$

Second-Preimage Resistance x'
For random x, k, hard to find $x' \neq x$ s.t. $h_k(x) = h_k(x')$

One-Way x'
For random x, k, hard to find x' s.t. $h_k(x) = h_k(x')$

Universality (x, x')
Hard to find $x \neq x'$ s.t. for random k, $h_k(x) = h_k(x')$

RSA Encryption
$N = pq$ where $p, q \in \mathbb{Z}_{prime}$
$pk = (N, e) \leftarrow \gcd\left(e, \emptyset(N)\right) = 1$
$sk = (N, d) \leftarrow d = e^{-1} (\mathrm{mod}\, \emptyset(N))$
- Not CPA-Secure
- Not CCA-Secure

Key Encapsulation Mechanism (KEM)
$Gen(\lambda) \to (pk, sk)$
$Encaps(\lambda, pk) \to (k, \hat{k})$
$Decaps(\lambda, sk, \hat{k}) \to k$

Trapdoor Permutations (TDP)
- One-way
  - Easy $a \to a^e (\mathrm{mod}\, N)$
- Invertible
  - Hard $a \leftarrow a^e (\mathrm{mod}\, N)$
  - Easy if given $d = e^{-1} (\mathrm{mod}\, \emptyset(N))$

Random Oracle Model (ROM)
Everyone has access to H
If x not seen before: $H(x)$ is random
If x seen before: $H(x)$ is previous output

Signatures
$Gen(\lambda) \to (sk, vk)$
$Sign(\lambda, sk, m) \to \sigma$
$Verify(\lambda, vk, m, \sigma) \to$ accept or reject
- Publicly verifiable
- Non-Repudiation

Zero-Knowledge Proofs
Prove without revealing any secrets
Use Interactive Proofs

Confidentiality
- Eve can only read bits on the channel
- Cannot learn m
- CPA-Secure

Integrity
- Eve can modify bits on the channel
- Bob knows if $m' \neq m$
- EUF-CMA-Secure

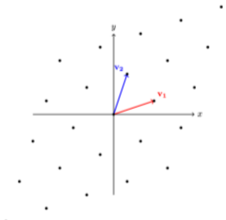
Authenticated Encryption (AE)
Must satisfy both confidentiality and integrity

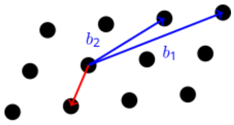
Non-Repudiation
Cannot deny ownership of the message

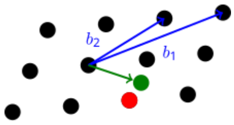
Group Theory
Identity	$\exists e \in G, \forall g \in G: e * g = g$
Inverse	$\forall g \in G, \exists g^{-1} \in G: g * g^{-1} = e$
Associativity	$\forall a, b, c \in G: (a * b) * c = a * (b * c)$
Exponentiation	$g * g * \ldots * g = g^x$

Bezout's Identity
- Let $\gcd(a, b) = d$, then $\exists x, y \in \mathbb{Z}: ax + by = d$
- $\forall c \in \mathbb{Z}: 0 < c < d, \nexists x, y \in \mathbb{Z}: ax + by = c$
- Suppose $\exists x: ax = 1 (\mathrm{mod}\, p)$
  - $ax = 1 + py, \forall y \in \mathbb{Z}$
  - $ax - py = 1$
  - $\gcd(a, p) = 1$

Euler's Theorem
$a^{\emptyset(N)} = 1 (\mathrm{mod}\, N)$, where $\emptyset(N) = (p-1)(q-1)$

Lattice Problems


Shortest Vector Problems: Hard to shortest vector


Closest Vector Problems: Hard to closest vector


Learning With Errors (LWE) Problem
Search LWE
Given $(A, As + e)$ hard to find s

Decision LWE
Hard to distinguish between $(A, As + e)$ and (A, b)

Quantum Encryption
Symmetric-Key	Public-Key
$Gen(n) \to s \in \mathbb{Z}_q^n$	$Gen(n) \to (A, As + e)$, $sk = s$ • $A \leftarrow \mathbb{Z}_q^{m \times n}$ • $e \leftarrow [-\eta q, \eta q]^m$ • $s \in \mathbb{Z}_q^n$
$Enc(s, \mu) \to$ $\left(a, \langle a, s \rangle + e + \mu \left\lfloor \frac{q}{2} \right\rceil\right)$ • $a \in \mathbb{Z}_q^n$ • $e \leftarrow [-\eta q, \eta q]$	$Enc(A, b, \mu) \to$ $\left(r^T A, r^T b + e\mu \left\lfloor \frac{q}{2} \right\rceil\right)$ • $r \in \{0,1\}^m$
$Dec(s, a, b) \to$ 1 or 0 • $z = b - \langle a, s \rangle$ $= \mu \lfloor q/2 \rceil + e$ • Output $\|z\| > \frac{q}{4}$	$Dec(s, c, d) \to$ 1 or 0 • $z = \langle c, s \rangle - d$ $= r^T As - r^T b - \mu \left\lfloor \frac{q}{2} \right\rceil$ $= r^T e - \mu \left\lfloor \frac{q}{2} \right\rceil$ $= -\mu \left\lfloor \frac{q}{2} \right\rceil$ • Output $\|z\| > \frac{q}{4}$

CI-Secure

1. Challenger C generates $k \leftarrow Gen(\lambda)$
2. C picks $b \in \{0,1\}$ uniformly at random
3. A sends $m_0, m_1 \in M, m_0, m_1 \notin \{m_i\}$ to C
4. C encrypts $Enc(k, m_b) \rightarrow c$ and sends c to A
5. A outputs b'. A wins iff $b' = b$

CPA-Secure CI under Chosen Plaintext attack (IND-CPA)
Encryption

1. Challenger C generates $k \leftarrow Gen(\lambda)$
2. C picks $b \in \{0,1\}$ uniformly at random
3. Adversary A can send multiple encryption queries $Enc(k, m_i) \rightarrow c_i$
4. A sends $m_0, m_1 \in M, m_0, m_1 \notin \{m_i\}$ to C
5. C encrypts $Enc(k, m_b) \rightarrow c$ and sends c to A
6. A outputs b'. A wins iff $b' = b$

PRF / PRP

1. Challenger C generates $k \leftarrow Gen(\lambda)$
2. C picks $b \in \{0,1\}$ uniformly at random
3. Adversary A can send multiple evaluation queries $Eval(k, x_i) \rightarrow y_i$
4. A sends $x \in \{0,1\}^\lambda$ to C
5. C sends $g(x) \rightarrow y, x \notin \{x_i\}$ to A
   - 5.1. If $b = 0, g = f_k$
   - 5.2. If $b = 1, g = f \in F_\lambda$
6. A outputs b'. A wins iff $b' = b$

EUF-Secure (Existential Unforgeability)

1. Challenger C generates $k \leftarrow Gen(\lambda)$
2. A sends $m \in M$ to C
3. C sends tag $t \leftarrow Mac(k, m)$ and to A
4. A outputs $(m', t'), m' \neq m$.
5. A wins iff $Verify(\lambda, k, m', t') \rightarrow$ accept

EUF-CMA-Secure (EUF under Chosen Message Attack)

1. Challenger C generates $k \leftarrow Gen(\lambda)$
2. Adversary A can send multiple MAC queries $Mac(k, m_i) \rightarrow t_i$
3. A outputs $(m', t'), m' \notin \{m_i\}$.
4. A wins iff $Verify(\lambda, k, m', t') \rightarrow$ accept

AE-Secure
Prove both CPA-Secure and EUF-CMA-Secure
Is CCA-Secure

CCA Security Chosen Ciphertext Attacks

1. Challenger C generates $k \leftarrow Gen(\lambda)$
2. C picks $b \in \{0,1\}$ uniformly at random
3. Adversary A can send multiple encryption queries $Enc(k, m_i) \rightarrow c_i$ and decryption queries $Dec(k, c_i) \rightarrow m_i'$
4. A sends $m_0, m_1 \in M, m_0, m_1 \notin \{m_i\}$ to C
5. C encrypts $Enc(k, m_b) \rightarrow c$ and sends c to A
6. A outputs b'. A wins iff $b' = b$

KE-Secure Key Exchange

1. Alice and Bob will pass σ_i to each other
2. Eve wins if can output k after seeing σ

KEI-Secure Indistinguishability for Keys (KE-IND)

1. Simulate a Key Exchange game and output σ
2. C picks $b \in \{0,1\}$ uniformly at random
3. C outputs σ, k
   - 3.1. If $b = 0, k \leftarrow Gen(\lambda)$
   - 3.2. If $b = 1, k \in K$
4. A outputs b. A wins iff $b' = b$

KEM-Secure Key Encapsulation Mechanism (KEM-CPA)

1. Challenger C generates $(pk, sk) \leftarrow Gen(\lambda)$
2. C generates $(k, \hat{k}) \leftarrow Encaps(\lambda, pk)$
3. C picks $b \in \{0,1\}$ uniformly at random
4. C outputs $(pk, k', \hat{k})$
   - 4.1. If $b = 0, k' = k$
   - 4.2. If $b = 1, k' \in K$
5. A outputs b. A wins iff $b' = b$

One-Time Pad
$Gen(\lambda) \rightarrow k \in \{0,1\}^n$
$Enc(\lambda, k, m) \rightarrow c = k \oplus m$
$Dec(\lambda, k, c) \rightarrow m' = k \oplus c$

- Limitation: cannot reuse keys
  - $c_0 \oplus c_1 = m_0 \oplus m_1$

ElGamal Encryption
$Gen(\lambda) \rightarrow (g, g^x)$
$Enc(\lambda, k, m) \rightarrow (g^y, g^{xy} \cdot m)$
$Dec(\lambda, k, c) \rightarrow (g^{xy})^{-1}(g^{xy} \cdot m)$

Discrete Log Problem
Given g and h, hard to find x: $g^x = h$

Discrete Log Assumption
$\forall$ PPT A $\exists$ negl $\mathcal{I}$:
$$P_{\substack{p \leftarrow PRIME_\lambda \\ g \leftarrow Gen_p \\ h \leftarrow \mathbb{Z}_p^* \\ x \leftarrow A(p,g,h)}}[h = g^x (\mathrm{mod}\, p)] < \mathcal{I}(\lambda)$$

Computational Diffie-Hellman (CDH) Assumption
$\forall$ PPT A $\exists$ negl $\mathcal{I}$:
$$P_{\substack{p \leftarrow PRIME_\lambda \\ g \leftarrow Gen_p \\ a,b \leftarrow \{0,1,\ldots,p-2\}}}[A(p, g, g^a, g^b) = g^{ab}] < \mathcal{I}(\lambda)$$

Decisional Diffie-Hellman (DDH) Assumption
Given g, g^a, g^b, then g^{ab} is hard to recognise

- Not true with $\mathbb{Z}_p^*$

RSA Assumption
$\forall$ PPT A $\exists$ negl $\mathcal{I}$:
$$P_{\substack{p,q \leftarrow PRIME_\lambda \\ N \leftarrow pq \\ e : \gcd(e, \emptyset(N)) = 1}}[A(N, e, a^e \bmod N) = a] < \mathcal{I}(\lambda)$$