Symmetric encryption: $Enc: K_\lambda \times M_\lambda \to C_\lambda$, $Dec: K_\lambda \times C_\lambda \to M_\lambda$

Correctness: $Dec_\lambda(k, Enc_\lambda(k, m)) = m$ always

Security: Distribution of all tuples of messages computationally indistinguishable from each other (and from random)

Negligible function: for every $c \in \mathbb{N}$, $\exists n_c \in \mathbb{N}$. $\forall n > n_c$, $u(n) < n^{-c}$

Pseudorandom function (PRF): $f_s(x)$. The distribution of functions uniformly over the family is computationally indistinguishable from the uniform distribution over all functions on that domain and codomain

CPA secure encryption from PRF $r \xleftarrow{R}$ randomness space

$Enc(k,m) = (r, m \oplus F(k,r))$, $Dec(k,(r,c)) = F(k,r) \oplus c$

PRF security game: adversary with polynomial queries cannot distinguish between PRF and actual random function.

In practice, keysize, num queries, and "negligible" probability are often fixed

If P=NP then PRFs do not exist.

CPA security game: In each step, adversary sends two messages of the same length. Challenger returns encryption of message corresponding to pre-selected bit. After polynomial steps the adversary has negligible advantage in guessing bit.

Counter mode:

$Enc(k, (m_1, \dots m_\ell))$: sample $r \xleftarrow{R} \{0,1\}^n$ r is a nonce

output $(r, F(k,r) \oplus m_1, F(k,r+1) \oplus m_2, \dots)$

$Dec(k, (r, c_1, \dots c_\ell))$: output $(F(k,r) \oplus c_1, F(k,r+1) \oplus c_2, \dots)$

This is a CPA secure encryption scheme, can be parallelized.

Pseudorandom permutation (PRP): $P, P^{-1}: K \times X \to X$

PRP consists of two efficient algorithms $X = \{0,1\}^n$

Maps distinct inputs to distinct outputs

Correctness: P and P^{-1} are inverses of each other

Security game: adversary with polynomial queries cannot distinguish between PRP and actual random permutation

PRP with counter mode encryption is secure, PRP is equally secure as PRF if T = queries, $T^2 \ll 2^n$.

To break when $T^2 \approx 2^n$, birthday collision attack

Even-Mansour cipher: $2n$ bit key, n bit block size, public π

$P_{EM}((k_0, k_1), x) := k_1 \oplus \pi(x \oplus k_0)$

AES: round keys $k_0, \dots, k_r \in \{0,1\}^n$ are linear funcs of input key k

$st \leftarrow x \oplus k_0$

For $i = 1, \dots, r$:

$st \leftarrow \pi(st) \oplus k_i$

output st

Public-key encryption:

Gen: PPT algorithm, takes 1^λ security parameter and outputs (pk, sk) key pair

Enc: PPT algorithm, takes pk and message m and outputs ciphertext $Enc(pk, m)$

Dec: poly time algorithm, takes sk and ciphertext ct and outputs message m

Correctness: $P[Dec(sk, Enc(pk, m)) = m] = 1$

Security: for all λ and all $m_0, m_1 \in M_\lambda$, $(pk, Enc(pk, m_0)) \approx (pk, Enc(pk, m_1))$

Learning with Errors (LWE assumption):

$A \xleftarrow{R} \mathbb{Z}_q^{m \times n}$ uniform, $s \xleftarrow{R} \mathbb{Z}_q^n$, $e \leftarrow \chi^m$ where χ is error dist, $\chi \in [-B, B]$, $B << q$

$\left[A\right] \cdot [s] + \left[e\right] \approx \left[u\right]$

Symmetric encryption:

To encrypt: Output $(A, A \cdot s + \lfloor q/2 \rfloor \cdot v)$ where A is random matrix and e is random error

To decrypt: compute $c = b - A \cdot s$, round each entry to multiple of $\lfloor q/2 \rfloor$, divide by $\lfloor q/2 \rfloor$, output result

LWE is linearly homomorphic with error propagation.

Private Information Retrieval (PIR): (Query, Answer, Reconstruct)

Server holds N-bit database, user wants ith bit without revealing i to the server.

- correctness: correctly recover D_i with probability $1 - negl$
- security: server's view is computationally indistinguishable between user queries for i or j
- succinctness: length of query and answer is at most N bits total

Square root PIR protocol: database $D \in \mathbb{Z}_2^{\sqrt{N} \times \sqrt{N}}$, user reads $(i,j) \in [\sqrt{N}] \times [\sqrt{N}]$

$Query(1^n, (i,j)) \to (qu \in \mathbb{Z}_q^{\sqrt{N} \times n} \times \mathbb{Z}_q^{\sqrt{N}}, st \in \mathbb{Z}_q^n \times [\sqrt{N}])$

- Build unit vector u_j of all 0s except for 1 at position j
- Sample random $A \in \mathbb{Z}_q^{\sqrt{N} \times n}$, $s \in \mathbb{Z}_q^n$, $e \in \chi^{\sqrt{N}}$
- Output query $qu \leftarrow (A, A \cdot s + e + \lfloor q/2 \rfloor \cdot u_j)$, state $st \leftarrow (s, i)$

$Answer(D, qu) \to ans \in \mathbb{Z}_q^{\sqrt{N} \times n} \times \mathbb{Z}_q^{\sqrt{N}}$:

- Parse query qu as (A, b), output answer $ans \leftarrow (D \cdot A, D \cdot b)$

$Reconstruct(ans, st) \to bit \in \{0,1\}$

- Parse state as (s, i), ans as (H, c), compute $v = c - H \cdot s \in \mathbb{Z}_q^{\sqrt{N}}$
- round each entry to multiple of $\lfloor q/2 \rfloor$, divide by $\lfloor q/2 \rfloor$, output result.

Optimize by using fixed A, precomputed $H \leftarrow D \cdot A$

Fully Homomorphic Encryption (FHE): build from LWE parameters, $\ell = (n+1)\log q$

$KeyGen(1^n) \to s \in \mathbb{Z}_q^{n+1}$: sample random s' from $\mathbb{Z}_q^n$, output $s = \binom{-s'}{1} \in \mathbb{Z}_q^{n+1}$ as secret key

$Enc(s \in \mathbb{Z}_q^{n+1}, u \in \mathbb{Z}_2) \to C \in \mathbb{Z}_q^{\ell \times (n+1)}$: Sample random $A \xleftarrow{R} \mathbb{Z}_q^{\ell \times n}$, error vector $e \xleftarrow{R} \chi^\ell$

- Build matrix $B = (A \,||\, As + e) \in \mathbb{Z}_q^{\ell \times (n+1)}$, let G be fixed error correcting matrix
- Output $C = B + u \cdot G$ as ciphertext

$Dec(s \in \mathbb{Z}_q^{n+1}, C \in \mathbb{Z}_q^{\ell \times (n+1)}) \to u \in \mathbb{Z}_2$: compute vector $v = C \cdot s$,

- output 0 if the magnitude of each entry of v is small ($< q/4$) otherwise 1

Homomorphic addition: add two matrices $C = C_1 + C_2$

Homomorphic multiplication: $C = h(C_1) \cdot C_2$ where h is inverse of G; $h(C) \cdot G = C$

$G = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ 2^{\log q - 1} \\ \hline & & & 1 \\ & & & 2 \\ & & & \vdots \\ & & & 2^{\log q - 1} \end{bmatrix}$

Adding doubles error bound.

Multiplying multiplies error by $\approx n \log q$

Can decrypt $q >> (n \log q)^\lambda \cdot 2B$, $d \approx n^{0.99}$

Full FHE: Use Dec to "refresh" and remove the error from the ciphertext

Can compute Dec homomorphically, $ek = (ct_{s_1}, \dots ct_{s_2})$, now error accumulates from circuit size

Problems:

encryption: does $Enc(s,b) \cdot s = b(-s, 1)$?

security: is the output matrix random?

homomorphic addition: is $(c_1 + c_2)s = (b_1 + b_2)s$?

homomorphic multiplication: is $C_1 C_2 s = b_1 b_2 s$?

C_1 AND C_2: $C_1 \times C_2$

C_1 OR C_2: $(C_1 + C_2) - (C_1 \times C_2)$

C_1 XOR C_2: $(C_1 + C_2) - 2(C_1 \times C_2)$

NOT C_1: $1 - C_1$

Interactive Proofs and Zero-knowledge: messages can depend on prev responses

Completeness: If true, accepted by verifier with probability $\geq \frac{2}{3}$

Soundness: If false, accepted by verifier with probability $\leq \frac{1}{3}$

ZKPs: prove knowledge of preimage under some function $f: X \to Y$ $f(x) = y$ without revealing x itself

Completeness: If the prover truly knows x, can always convince verifier

Knowledge soundness: A cheating prover can only convince verifier if prover actually "knows" a preimage x such that $f(x) = y$

Zero-knowledge: The verifier learns nothing about x from the interaction, besides $y = f(x)$

Signature scheme: $vk = A \cdot s + e$, $sk = (A, b, s, e)$, $vk = (A, b)$

$Sign(sk, m) \to \sigma$. Prove knowledge of a solution to LWE instance (A, b) using sk

$Ver(vk, m, \sigma) \to \{0,1\}$ Run verifier, accept iff verifier accepts

Commitment scheme: $Gen(1^\lambda)$ outputs public parameter $pp \in \{0,1\}^N$, $N \in poly(\lambda)$

$Com(pp, m, r)$ outputs commitment, where randomness $r \xleftarrow{R} \{0,1\}^\lambda$

Hiding: $\forall m_0, m_1 \in M$, $Com(pp, m_0, r_0) \approx Com(pp, m_1, r_1)$

commitments are indistinguishable (usually computationally)

Binding: $P[Com(pp, m_0, r_0) = Com(pp, m_1, r_1)$ and $m_0 \neq m_1] = negl(\lambda)$

cannot find two messages that yield the same commitment (computational/statistical)

Construction from LWE: $Gen(1^\lambda)$ chooses $A \xleftarrow{R} \mathbb{Z}_q^{m \times n}$, $u \xleftarrow{R} \mathbb{Z}_q^m$,

outputs $pp = (A, u)$. $Com((A,u), b, (s,e)) = As + e + bu$

Secret sharing: each person gets a share, any t out of n can reconstruct secret

Coalitions of less than t should have no information about distribution of secret.

Shamir scheme: t out of n, message m

Choose random degree $t-1$ polynomial f in $\mathbb{Z}_p$ (with $p > n$) such that $f(0) = m$

For every $i \in [n]$ let $s_i = f(i)$.

To reconstruct, interpolate by solving t linear eqs in t variables, to recover polynomial

Multi-party computation BGW Protocol consists of three phases:

1. secret sharing of inputs: each party shares input using Shamir scheme.
2. parties compute f on their shares, where f is represented by additions and multiplications $g_{a+b}(i) = g_a(i) + g_b(i)$, $g_{c \cdot a}(i) = c \cdot g_a(i)$, $g_{a \cdot b}(i) = g_a(i) \cdot g_b(i)$ then degree reduction and rerandomize. deg reduction: take equivalent matrix, convert to polynomial rerandomization: each party adds to its share the ith share of $g_{a \cdot b} + g_1 + \dots + g_n$
3. Shamir scheme is the Reed-Solomon code of $(f(0), f(1), \dots, f(t))$ of length $t+1 = k$ which can be decoded if at most $\frac{n-k}{2} = \frac{n-t-1}{2}$ coordinates are corrupted

One-Way function: family of functions $\{f_\lambda\}_{\lambda \in \mathbb{N}}$, $f_\lambda: \{0,1\}^\lambda \to \{0,1\}^*$

$$P\left[f_\lambda(x) = f_\lambda(x') : \begin{matrix} x \leftarrow \{0,1\}^\lambda \\ x' \leftarrow A(f_\lambda(x)) \end{matrix}\right] \leq negl(\lambda)$$

Given $f_\lambda(x)$ and x, it is difficult to find x' such that $f_\lambda(x') = f_\lambda(x)$

Collision resistant: family of functions $\{f_\lambda\}_{\lambda \in \mathbb{N}}$, $f_\lambda: \{0,1\}^* \to \{0,1\}^\lambda$

$$P\left[f_\lambda(x) = f_\lambda(x') \text{ and } x \neq x' : (x, x') \leftarrow A(1^\lambda)\right] \leq negl(\lambda)$$

It is difficult to find distinct x, x' such that $f_\lambda(x) = f_\lambda(x')$