Lecture 6

Sunday, September 15, 2024 12:22 PM

Ref. PP computing by Chen Ch. 3

Homomorpic Encryption (HE)

Def^n

An HE solution consists of four components: keyGen, Enc, Dec, and Evaluation

- keyGen(1) → (Pk, Sk): Given ecryption parameter λ, it generates pair Pk, Sk
- Enc(Pt, Pk) → Ct: Given Pk, it encrypts a plaintext Pt into ciphertext Ct
- Dec(Sk, Ct) → Pt: Given Sk, it decrypts the ciphertext Ct into Pt
- Evaluate(Pk, Π, Ct₁, Ct₂, ...) → (Ct̀₁, Ct̀₂, ...): Given the Pk, the input ciphertexts (Ct₁, Ct₂, ...) and the computing function Π, it will perf Π on Ct₁, Ct₂, ... a produces Ct̀₁, Ct̀₂, ...

Def^n (Correctnes)

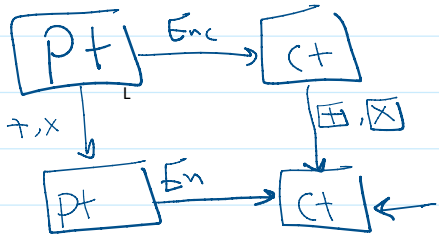
Generally HE is correct for operation Π if it correctly decrypts ciphertexts with the following prop.

1- HE is correG if it always retrives a Pt that has not been evaluated

→ $Pr[Dec(sk, Enc(Pk, Pt)) = Pt] = 1$

2- HE correctly decrypts ciphertexts evaluated on Π

$$Pr[Dec(Sk, Evaluate(Pk, \Pi, Ct_1, Ct_2, \dots)) = \Pi(Pt_1, Pt_2, \dots)] = 1$$



Ex

RSA

Recall Pk (e, n) and Sk(d)

$Enc(X) = X^e \bmod n = Ct_1$

$Enc(Y) = y^e \bmod n = Ct_2$

$Evaluate(Pk, X, Ct_1, Ct_2) = X^e \bmod n \times y^e \bmod n = (X \cdot y)^e \bmod n = Enc(X \cdot y)$

$dec(Enc(x \cdot y)) = (X \cdot Y)$