Tutorial 10 (TA. John Ehab)

- Homomorphism allows special operations to be performed on encrypted data without decrypting it first, and the results of these computations remain encrypted.

Enhancing Privacy, no one can see neither your data, nor the output of the computations done on it.

Use cases: Data Analysis, Images Filtering, DNA Tests, …

- Operations are ADDITION and MULTIPLICATION.

Computations are done using a public evaluation key (could be different than the public key used for encryption).

- HE is usually represented by arithmetic circuits.

Size of the circuit: no. of gates / operations.

Depth: the longest path between an input and an output.

- The security of HE relies on the hardness of the Learning With Errors (LWE) problem, even for quantum computers, because it's lattice-based.

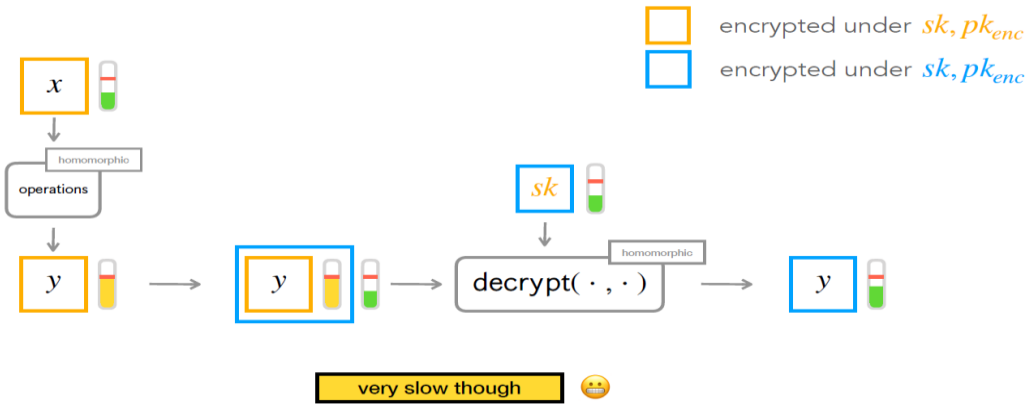
Read more about LWE.

- "Noises" are added not to make it easy to reverse the process, produce a unique output every time. Too much operations will lead to too much noise.

To solve this, we use "bootstrapping".

Bootstrapping is an extra step of encrypting and decrypting to remove some noise.

Solution - Bootstrapping



- Homomorphic Encryption has 3 types:

    1. Partially-HE (PHE): only ONE type of homomorphic operation, either addition or multiplication, but not both.

    2. Leveled-HE (LHE): combination of both types accepted, but a LIMITED number of homomorphic operations.

    3. Fully-HE (FHE): Unlimited number of operations. We used bootstrapping to go from LHE -> FHE.

- Homomorphic Encryption is a groundbreaking and relatively new area in cryptography. While it holds incredible potential for enabling secure computations on encrypted data, it is still HIGHLY computationally expensive and inefficient.
Ongoing research is actively working to improve its efficiency and reliability for more practical applications.
The research world is wide open for your contribution!😊