ð§® FHE Schemes Overview

Fully Homomorphic Encryption (FHE) schemes are divided into three generations, each designed for different types of applications.

Each of these generations relies on solving complex problems like Learning with Errors (LWE) and its generalization Ring LWE (RLWE) to ensure security.

We believe that understanding the advantages and disadvantages of each scheme will be important in being able to provide developers with the right tool for the application that they are trying to create.

First Generation - Integer Arithmeticâ

BGV Schemeâ

The BGV scheme was the first practical, leveled homomorphic encryption method. It introduced a technique called "packing," allowing multiple plaintexts to be encrypted into a single ciphertext, making it efficient in handling multiple data points simultaneously (like SIMD in processors). It avoided the need for bootstrapping, although it also included a bootstrapping option to upgrade to a fully homomorphic scheme.

Second Generation - Binary Operationsâ

GSW Schemeâ

The GSW scheme introduces a unique approach for performing homomorphic operations called the "approximate eigenvector method." This method eliminates the need for "modulus switching" and "key switching." Instead, it uses multiplication via tensoring, which is later formalized as using a "gadget matrix." This approach significantly reduces error growth, but it does result in larger ciphertexts and higher computational costs. Due to these drawbacks, computations are limited to a binary message space. There is also an RLWE version of this scheme.

FHEW Schemeâ

The FHEW scheme is an optimized version of the GSW scheme, focusing on bootstrapping efficiency. It treats decryption as an arithmetic function rather than a boolean circuit. This RLWE variant incorporates several optimizations, making GSW-based bootstrapping faster than the BGV scheme. Key improvements include:

1. Restricting computations to a binary message space and using a NAND gate for homomorphic operations.
2. Enabling the evaluation of arbitrary functions via lookup tables during bootstrapping, known as âprogrammable bootstrapping.â
3. Utilizing efficient Fast Fourier Transform (FFT) methods for faster computations.

TFHE Schemeâ

This scheme uses "Blind Rotation" to enable fast bootstrapping, which is the process of refreshing a ciphertext to prevent error accumulation from making it unusable. It involves two layers of encryption: a basic Learning with Errors (LWE) encryption and a special ring-based encryption for secure and efficient computation. The TFHE scheme builds on FHEW techniques and employs methods like "modulus switching" and "key switching" for improved performance.

Third Generation - Approximate Number Arithmeticâ

CKKS Schemeâ

CKKS introduces an innovative way to map real (or complex) numbers for encryption. It includes a "rescaling" technique to manage noise during homomorphic computations, reducing ciphertext size while preserving most of the precision. Originally a leveled scheme, it later incorporated efficient bootstrapping to become fully homomorphic and added support for packed ciphertexts. Edit this page

Previous ð¤² 3rd party Integrations Next Details