**Security Analysis**

This is security analysis of “**Bank Finance Calculator**” application which showcases the practical application of Homomorphic Encryption using Microsoft Seal Library.

The CKKS (Cheon-Kim-Kim-Song) homomorphic encryption scheme is designed to perform operations on encrypted real or complex numbers, making it suitable for privacy-preserving computations involving numerical data. Below is the security analysis of “Bank Finance Calculator” application which used CKKS scheme for homomorphic encryption.:

**Semantic Security:**

CKKS is based on the **Learning With Errors (LWE)** assumption, which is a well-studied and widely accepted assumption in lattice-based cryptography.

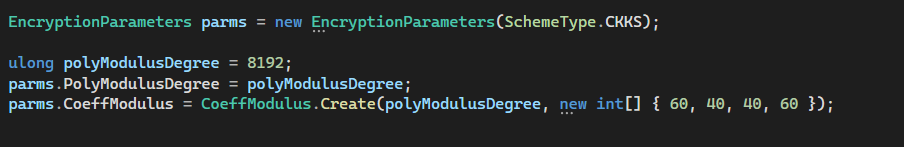
**Security:**

Our implementation sets the initial scale to 2^40. Choosing an appropriate scale is essential to avoid overflow or underflow during computations and maintain the security of the scheme.



**Parameter Selection:**

Proper parameter selection, such as choosing suitable values for the polynomial modulus degree, coefficient modulus, and scale, is critical for security. Our implementation uses parameters which are carefully chosen based on the security requirements and performance considerations.



**Key Management:**

Key generation, distribution, and management are fundamental aspects of any cryptographic system. As this is demo application, so keys are generated during execution of application for encryption and decryption processes.

**Homomorphic Operations:**

CKKS supports addition, multiplication, and other homomorphic operations. These operations are performed carefully to avoid leakage of information through side-channel attacks or improper handling of noise.

**Scale Setting:**

2^40 is appropriate scale to deal with dynamic range of the data accurately and efficiently.

**Testing and Validation:**

Comprehensive testing and validation is done to ensure the correctness and security of the implementation.