PRE week5

▼ 📃 **week5 task**

- OpenFHE의 Cipertext를 class로 만들어서 scale 및 decryption vector 추적
  ex. Cipertext ct
  ct.showDetail

> scale : ~~
> decryption : ~~
> original : ~~
> 암호화한 시점에 보여주고, 연산할수록 얼마나 차이가 나는지 확인할 수 있도록 한다.

📌 코드 분석

▼ `ciphertext-fwd.h`

```
// Input
std::vector<double> x = {1.0, 1.01, 1.02, 1.03, 1.04, 1.05, 1.06, 1.07};
Plaintext ptxt        = cc->MakeCKKSPackedPlaintext(x);

std::cout << "Input x: " << ptxt << std::endl;

auto c = cc->Encrypt(ptxt, keys.publicKey);
```

- 암호문을 담을 때 사용되는 auto의 정의로 이동 → ciphertext-fwd.h

```
/*
 * It is a lightweight file to be included where we need the declaration of Ciphertext only
 */

#ifndef __CIPHERTEXT_FWD_H__
```

```
#define __CIPHERTEXT_FWD_H__

#include <memory>

namespace lbcrypto {

template <typename Element>
class CiphertextImpl;

template <typename Element>
using Ciphertext = std::shared_ptr<CiphertextImpl<Element>>;

template <typename Element>
using ConstCiphertext = std::shared_ptr<const CiphertextImpl<Element>>;

}  // namespace lbcrypto

#endif  // __CIPHERTEXT_FWD_H__
```

- `ciphertext-fwd.h` : Ciphertext의 선언에 쓰이는 가벼운 파일
- `CryptoContext` : 암호화된 데이터의 컨텍스트를 나타내는 class
- `CryptoContextImpl` : CryptoContext 클래스의 구현. 실제로 CryptoContext의 기능과 동작을 구현하는 부분

▼ **`ciphertext.h`**

- `CiphertextImpl` 클래스 : 암호문을 포함하는 Ciphertext 객체. CryptoObject<Element> 클래스를 상속받는다.

▼ `CiphertextImpl` 의 멤버 변수들

- `m_elements` : Ciphertext의 요소를 나타내는 Element 클래스의 벡터. 암호화 된 데이터가 여러 개의 요소로 구성된 경우 사용됩니다.
- `m_noiseScaleDeg` : 암호화된 메시지의 스케일링 인수(degree)를 나타냅니다. 이 값은 노이즈 처리에 사용되는 매개변수
- `m_level, m_hopslevel`: Ciphertext의 레벨과 재암호화 레벨

▼ `CiphertextImpl` 의 멤버 함수들

- `GetElement(), GetElements(), SetElement(), SetElements()`: Ciphertext의 요소에 접근하고 설정하는 함수들
- `GetScalingFactor()` : Ciphertext의 스케일링 팩터에 접근하는 함수
- `GetScalingFactorInt()` : Ciphertext의 스케일링 팩터(정수 값)에 접근하는 함수
-

📌 궁금한 점

- scale과 scaling factor의 차이는 무엇인지 궁금합니다.