openFHE에서 암호문을 만들었을 때, 자동으로 로그가 찍히도록 하는 새로운 클래스 TraceCipherText를 구현.

1. 스케일 표시
2. 레벨 표시
3. 복호문 표현

상속을 통해 구현하려 하였으나, 아직 Ciphertext 타입의 구조를 잘 몰라 상속시 오류가 많이 났다.
그래서 우선 상속이 아닌 새로운 클래스를 구현하여 그 클래스 안에 ciphertext, CyptoContext, PrivateKey를 가지고 있도록 구현하였다.

구현하기에는 편했지만, 클래스를 만들 때 마다 세가지 요소를 모두 넣어줘야하고, 이전에 쓰던 형식과 다르게 사용해야하기 때문에 상속을 통해 더 편리하게 구현할 계획이다.

```
class TraceCipherText{
    private:
        Ciphertext<DCRTPoly> ct;
        CryptoContext<DCRTPoly> cc;
        PrivateKey<DCRTPoly> pk;
    public:
        TraceCipherText(Ciphertext<DCRTPoly> a, CryptoContext<DCRTPoly> cc, PrivateKey<DCRTPoly> pk) : ct(a), cc(cc), pk(pk) {
            showDetatil();
        }
        void showDetatil(){
            cout << "decode text : ";
            decode();
            cout << "scale : " << ct->GetScalingFactor() << endl;
            cout << "Level : " << ct->GetLevel() << endl;
            cout << endl;
        }
        Ciphertext<DCRTPoly> ciphertext(){
            return ct;
        }

        void decode(){
            Plaintext plaintext;
            cc->Decrypt(pk, ct, &plaintext);
            plaintext->SetLength(8);
            cout << plaintext;
        }
```

```
    void cal_error(Plaintext plaintext){
        pair<double, int> max_error = make_pair(-1, 0);
        pair<double, int> min_error = make_pair(1000, 0);
        for(int i = 0; i < trace.size(); i++){
           if(abs(trace[i] - (double)plaintext->GetRealPackedValue()[i]) > max_error.first) {
                max_error.first = abs(trace[i] - (double)plaintext->GetRealPackedValue()[i]);
                max_error.second = i;
            }
            if(abs(trace[i] - (double)plaintext->GetRealPackedValue()[i]) < min_error.first) {
                min_error.first = abs(trace[i] - (double)plaintext->GetRealPackedValue()[i]);
                min_error.second = i;
                //cout << trace[i] - (double)plaintext->GetRealPackedValue()[i] << endl;
            }
        }
        cout << "high gap : " << max_error.first << " Cal value " <<
        (double)plaintext->GetRealPackedValue()[max_error.second] <<
        " True value : " << trace[max_error.second] << " index :  " << max_error.second << endl;

        cout << " low gap : " << min_error.first << " Cal value " <<
        (double)plaintext->GetRealPackedValue()[min_error.second]<<
        " True value : " << trace[min_error.second] << " index :  " << min_error.second << endl;
    }
};
```

ShowDetail 을 생성자로하여 현재 암호문의 정보를 보여준다.

```
================ TRACE TEST ================
decode text : (1, 1.01, 1.02, 2000, 1.04, 1.05, 1.06, 1.07,  ... ); Estimated precision: 41 bits
scale : 8.8544484e+20
Level : 0

decode text : (1, 1.0201, 1.0404, 4000000, 1.0816, 1.1025, 1.1236, 1.1449,  ... ); Estimated precision: 29 bits
scale : 1.2676506e+30
Level : 1

decode text : (1, 1.0201, 1.0404, 4000000, 1.0816, 1.1025, 1.1236, 1.1449,  ... ); Estimated precision: 29 bits
scale : 1.2676506e+30
Level : 1

decode text : (3, 3.0201, 3.0404, 4000002, 3.0816, 3.1025, 3.1236, 3.1449,  ... ); Estimated precision: 29 bits
scale : 1.2676506e+30
Level : 1

decode text : (3, 3.0201, 3.0404, 4000002, 3.0816, 3.1025, 3.1236, 3.1449,  ... ); Estimated precision: 29 bits
scale : 1.2676506e+30
Level : 1

decode text : (4, 4.0401, 4.0804, 4004001, 4.1616, 4.2025, 4.2436, 4.2849,  ... ); Estimated precision: 29 bits
scale : 1.2676506e+30
Level : 1

decode text : (4, 4.0401, 4.0804, 4004001, 4.1616, 4.2025, 4.2436, 4.2849,  ... ); Estimated precision: 29 bits
scale : 1.2676506e+30
Level : 1

decode text : (11.998584, 12.206319, 12.404148, 1.6016012e+13, 12.828238, 13.047587, 13.252429, 13.472656,  ... ); Estimated precision: 6 b
its
scale : 1.2676506e+30
Level : 2

decode text : (12.003145, 12.191375, 12.415127, 1.6016012e+13, 12.829169, 13.041047, 13.260372, 13.47168,  ... ); Estimated precision: 6 bi
ts
scale : 1.2676506e+30
Level : 2


(x + 1)^2 * (x^2 + 2) : Result (11.999526, 12.202535, 12.406265, 1.6016012e+13, 12.823087, 13.014262, 13.246708, 13.484375,  ... ); Estimat
ed precision: 6 bits

high gap : 0.023994709 Cal value 13.014262 True value : 13.038256 index :  5
 low gap : 0.00021707976 Cal value 12.406265 True value : 12.406048 index :  2
```