REPORT

CBC CTR 성능 비교

과목명 :	컴퓨터보안
담당교수 :	정준호 교수님
제출일 :	2021 년 04 월 13 일
공과 대학	컴퓨터공학 과
학번 : 2016112154	이름: 정동구

1. CBC

```
public:
    CBC(std::string input)
    {
        input_string = input;
        for (int i = 0; i < 16; i++)
        {
            iv+=arr[rand()%16];
        }

    }
    void parseString()
    {
        int length = input_string.length();
        int block_num = length / 16;
        last_block_length = length % 16;
        /*if (length % 16 == 0)block_num = length / 16;
        else block_num = (length / 16) + 1;*/

        for (int i = 0; i < block_num; i++)
        {
            str_vec.push_back(input_string.substr(i * 16, 16));
        }
        if (length % 16 != 0)
        {
            str_vec.push_back(input_string.substr(16*block_num, length % 16));
            for (int j = 0; j < 16 -last_block_length; j++)
            {
                str_vec.back() += arr[rand() % 16];
            }
            //std::cout <<"₩n₩n"<< str_vec.back()<<"₩n₩n";
        }
    }
```

CBC class 의 생성자에서는 initialization vector 를 생성한다.
parseString 에서는 입력받은 문자열(16 진수)을 16 자씩 블록으로 나눈다. 마지막 블록이 16 자보다 짧을 경우 빈공간을 랜덤한 문자열로 채운다.

```
std::string XORBlock(std::string prev_enc_,std::string block_)
{
    std::vector<int> prev_enc=string2hex(prev_enc_);
    std::vector<int> block=string2hex(block_);
    std::vector<int> xor_str(prev_enc.size());
    for (int i = 0; i < prev_enc.size(); i++)
    {
        xor_str[i] = prev_enc[i] ^ block[i];
    }
    return hex2string(xor_str);

}
```

XOR 연산을 하는 함수이다.

```
void encrypt()
{
    parseString();
    std::string temp;

    for (int i = 0; i < str_vec.size(); i++)
    {
        if (i == 0)
        {
            DES des1(XORBlock(iv,str_vec[i]), "85E813540F0AB405", MODE::ENCRYPTION);
            enc_vec.push_back(des1.encryption());
            temp = XORBlock(enc_vec.back(),str_vec[i + 1]);
        }
        else
        {
            DES des1(temp, "85E813540F0AB405", MODE::ENCRYPTION);
            enc_vec.push_back(des1.encryption());
            if(i<str_vec.size()-1)temp = XORBlock(enc_vec.back(), str_vec[i + 1]);
        }
    }
}
```

암호화를 하는 함수이다. 첫 블록은 initialization vector 와 XOR 연산을 진행한다. 이후 블록부터는 이전 블록을 암호화 한 것과 XOR 연산을 진행한다.

```
void decrypt()
{

    std::string temp;

    for (int i = 0; i < enc_vec.size(); i++)
    {
        if (i == 0)
        {
            DES des1(enc_vec[i], "85E813540F0AB405", MODE::DECRYPTION);
            dec_vec.push_back(XORBlock(des1.decryption(),iv));
        }
        else
        {
            DES des1(enc_vec[i], "85E813540F0AB405", MODE::DECRYPTION);
            dec_vec.push_back(XORBlock(des1.decryption(),enc_vec[i-1]));
        }
    }
    if (last_block_length != 0)
    {
        dec_vec.back() = dec_vec.back().substr(0,last_block_length);
    }

}
```

복호화를 하는 함수이다. 첫 블록은 암호문을 복호화 후 암호화와 마찬가지로 initialization vector 와 XOR 연산을 하고 이후 블록부터는 복호화 후 이전 블록의 암호문과 XOR 연산을 한다.

2. CTR

```
class CTR
{
public:
    CTR(std::string input)
    {
        input_string = input;
        for (int i = 0; i < 8; i++)
        {
            nonce += arr[rand() % 16];
        }

    }
```

CTR 클래스의 생성자이다. Counter 를 생성하기 위한 nonce 를 만든다.

```
std::string counterStream(int index)
{
    std::string counter = nonce;
    std::string hex = dec2hex(index);
    for (int i = 0; i < 8 - hex.length(); i++)
    {
        counter += "0";
    }
    counter += hex;
    return counter;
}
```

Counter 를 생성하는 함수이다. 총 16 자리(64 비트)의 counter 를 생성하기 위해 8 자리의 nonce 에 8 자리의 16 진수를 이어붙인다. 16 진수는 블록의 순서와 동일하다.

```
void encrypt()
{
    parseString();
    for (int i = 0; i < str_vec.size(); i++)
    {
            DES des1(counterStream(i), "85E813540F0AB405", MODE::ENCRYPTION);
            enc_vec.push_back(XORBlock(des1.encryption(),str_vec[i]));
    }

}
void decrypt()
{

    std::string temp;

    for (int i = 0; i < enc_vec.size(); i++)
    {
            DES des1(counterStream(i), "85E813540F0AB405", MODE::ENCRYPTION);
            dec_vec.push_back(XORBlock(des1.encryption(),enc_vec[i]));
    }
    if (last_block_length != 0)
    {
        dec_vec.back() = dec_vec.back().substr(0, last_block_length);
    }
}
```

암호화와 복호화를 위한 함수이다. CTR 함수는 블록 암호를 스트림암호화 한 것으로 암호화와 복호화 과정이 동일하다.

```
void encrypt(int n)
{
    parseString();
    enc_vec.resize(str_vec.size());
    for (int i = n*(str_vec.size()/4); i < (n+1) * (str_vec.size() / 4); i++)
    {
            DES des1(counterStream(i), "85E813540F0AB405", MODE::ENCRYPTION);
            enc_vec[i] = XORBlock(des1.encryption(), str_vec[i]);
    }
}
```

병렬처리를 하기 위해 변경한 암호화 함수.

```
start = clock();
std::thread t1(&CTR::encrypt,CTR(plain_text),0);
std::thread t2(&CTR::encrypt, CTR(plain_text), 1);
std::thread t3(&CTR::encrypt, CTR(plain_text), 2);
std::thread t4(&CTR::encrypt, CTR(plain_text), 3);

t1.join();
t2.join();
t3.join();
t4.join();
end = clock();
result = (double)(end - start);
std::cout << "CTR encryption elapse time : " << result/ CLOCKS_PER_SEC << "₩n";
```

병렬처리를 한 부분.

총 4 개의 스레드로 멀티스레딩 하였다.

3. 암호화/복호화 결과

```
plain text: 0123456789ABCDEFFEDCBA98765432100A1B2C3D4E5F6
encryption


0A1B2C3D4E5F6CF3

Plain Text : [49D3DE5117BE3C3B]
   Key     : [85E813540F0AB405]
round     left          right         key
    1  [26E0E5B6]  [46C3AFA0]  [C842107B226C]
    2  [46C3AFA0]  [31DC22B8]  [35602242C637]
    3  [31DC22B8]  [E77BD01A]  [A205801F2D8C]
    4  [E77BD01A]  [303CB48A]  [580215A851D1]
    5  [303CB48A]  [5B1B5532]  [05901843E227]
    6  [5B1B5532]  [BF977338]  [0600E6F60D88]
    7  [BF977338]  [EC9002FA]  [BA480088135F]
    8  [EC9002FA]  [8C091290]  [08232857F2A0]
    9  [8C091290]  [2C1E36E0]  [23A0C408EA4F]
   10  [2C1E36E0]  [BF9BC88F]  [18448276D494]
   11  [BF9BC88F]  [91E472E7]  [7001388905EB]
   12  [91E472E7]  [FFA2071B]  [8480058EFA01]
   13  [FFA2071B]  [C82AFE51]  [030A16724774]
   14  [C82AFE51]  [6812B2B3]  [2C30A099898A]
   15  [6812B2B3]  [FA9E3DC8]  [920468C47611]
   16  [14B8B7EE]  [FA9E3DC8]  [0088D3B998C1]
Cipher Text : [0CA56DBBFC9D83B7]
Plain Text : [F279D7238AC9B1A7]
   Key     : [85E813540F0AB405]
round     left          right         key
    1  [F5CB329D]  [FFA08A65]  [C842107B226C]
    2  [FFA08A65]  [360FCA78]  [35602242C637]
    3  [360FCA78]  [3E35E58A]  [A205801F2D8C]
    4  [3E35E58A]  [4B8AD0D9]  [580215A851D1]
    5  [4B8AD0D9]  [2DB66138]  [05901843E227]
    6  [2DB66138]  [537D0D66]  [0600E6F60D88]
    7  [537D0D66]  [1C468E9C]  [BA480088135F]
    8  [1C468E9C]  [0759404E]  [08232857F2A0]
    9  [0759404E]  [2F0A95EC]  [23A0C408EA4F]
   10  [2F0A95EC]  [39EAFF00]  [18448276D494]
   11  [39EAFF00]  [C13C20BB]  [7001388905EB]
   12  [C13C20BB]  [511697A3]  [8480058EFA01]
   13  [511697A3]  [177AF207]  [030A16724774]
   14  [177AF207]  [4B623632]  [2C30A099898A]
   15  [4B623632]  [FC3ACA19]  [920468C47611]
   16  [78E1D600]  [FC3ACA19]  [0088D3B998C1]
Cipher Text : [122C84EAE6F0DC9C]
Plain Text : [1837A8D7A8AFB06F]
   Key     : [85E813540F0AB405]
round     left          right         key
    1  [7CF6B5AA]  [E6042D1F]  [C842107B226C]
    2  [E6042D1F]  [CEA43045]  [35602242C637]
    3  [CEA43045]  [BF880687]  [A205801F2D8C]
    4  [BF880687]  [7B7ADB22]  [580215A851D1]
    5  [7B7ADB22]  [491D3A43]  [05901843E227]
    6  [491D3A43]  [7FD36620]  [0600E6F60D88]
    7  [7FD36620]  [5982D8C1]  [BA480088135F]
    8  [5982D8C1]  [A7CCBFE5]  [08232857F2A0]
    9  [A7CCBFE5]  [22F62917]  [23A0C408EA4F]
   10  [22F62917]  [52CBC2AA]  [18448276D494]
   11  [52CBC2AA]  [7C7D845A]  [7001388905EB]
   12  [7C7D845A]  [484B9DB5]  [8480058EFA01]
   13  [484B9DB5]  [C96B1C3A]  [030A16724774]
   14  [C96B1C3A]  [5490A361]  [2C30A099898A]
   15  [5490A361]  [9C96400E]  [920468C47611]
   16  [3DC97181]  [9C96400E]  [0088D3B998C1]
Cipher Text : [5522E2D2E4441CB1]
0CA56DBBFC9D83B7 122C84EAE6F0DC9C 5522E2D2E4441CB1
```

```
decryption
Cipher Text : [0CA56DBBFC9D83B7]
    Key     : [85E813540F0AB405]
round     left          right         key
    1  [FA9E3DC8]  [6812B2B3]  [0088D3B998C1]
    2  [6812B2B3]  [C82AFE51]  [920468C47611]
    3  [C82AFE51]  [FFA2071B]  [2C30A099898A]
    4  [FFA2071B]  [91E472E7]  [030A16724774]
    5  [91E472E7]  [BF9BC88F]  [8480058EFA01]
    6  [BF9BC88F]  [2C1E36E0]  [7001388905EB]
    7  [2C1E36E0]  [8C091290]  [18448276D494]
    8  [8C091290]  [EC9002FA]  [23A0C408EA4F]
    9  [EC9002FA]  [BF977338]  [08232857F2A0]
   10  [BF977338]  [5B1B5532]  [BA480088135F]
   11  [5B1B5532]  [303CB48A]  [0600E6F60D88]
   12  [303CB48A]  [E77BD01A]  [05901843E227]
   13  [E77BD01A]  [31DC22B8]  [580215A851D1]
   14  [31DC22B8]  [46C3AFA0]  [A205801F2D8C]
   15  [46C3AFA0]  [26E0E5B6]  [35602242C637]
   16  [0FFE749B]  [26E0E5B6]  [C842107B226C]
Plain Text : [49D3DE5117BE3C3B]
Cipher Text : [122C84EAE6F0DC9C]
    Key     : [85E813540F0AB405]
round     left          right         key
    1  [FC3ACA19]  [4B623632]  [0088D3B998C1]
    2  [4B623632]  [177AF207]  [920468C47611]
    3  [177AF207]  [511697A3]  [2C30A099898A]
    4  [511697A3]  [C13C20BB]  [030A16724774]
    5  [C13C20BB]  [39EAFF00]  [8480058EFA01]
    6  [39EAFF00]  [2F0A95EC]  [7001388905EB]
    7  [2F0A95EC]  [0759404E]  [18448276D494]
    8  [0759404E]  [1C468E9C]  [23A0C408EA4F]
    9  [1C468E9C]  [537D0D66]  [08232857F2A0]
   10  [537D0D66]  [2DB66138]  [BA480088135F]
   11  [2DB66138]  [4B8AD0D9]  [0600E6F60D88]
   12  [4B8AD0D9]  [3E35E58A]  [05901843E227]
   13  [3E35E58A]  [360FCA78]  [580215A851D1]
   14  [360FCA78]  [FFA08A65]  [A205801F2D8C]
   15  [FFA08A65]  [F5CB329D]  [35602242C637]
   16  [274784EE]  [F5CB329D]  [C842107B226C]
Plain Text : [F279D7238AC9B1A7]
Cipher Text : [5522E2D2E4441CB1]
    Key     : [85E813540F0AB405]
round     left          right         key
    1  [9C96400E]  [5490A361]  [0088D3B998C1]
    2  [5490A361]  [C96B1C3A]  [920468C47611]
    3  [C96B1C3A]  [484B9DB5]  [2C30A099898A]
    4  [484B9DB5]  [7C7D845A]  [030A16724774]
    5  [7C7D845A]  [52CBC2AA]  [8480058EFA01]
    6  [52CBC2AA]  [22F62917]  [7001388905EB]
    7  [22F62917]  [A7CCBFE5]  [18448276D494]
    8  [A7CCBFE5]  [5982D8C1]  [23A0C408EA4F]
    9  [5982D8C1]  [7FD36620]  [08232857F2A0]
   10  [7FD36620]  [491D3A43]  [BA480088135F]
   11  [491D3A43]  [7B7ADB22]  [0600E6F60D88]
   12  [7B7ADB22]  [BF880687]  [05901843E227]
   13  [BF880687]  [CEA43045]  [580215A851D1]
   14  [CEA43045]  [E604201F]  [A205801F2D8C]
   15  [E604201F]  [7CF6B5AA]  [35602242C637]
   16  [884BAAAA]  [7CF6B5AA]  [C842107B226C]
Plain Text : [1837A8D7A8AFB06F]
0123456789ABCDEF FEDCBA9876543210 0A1B2C3D4E5F6
```

CBC 의 암호화 복호화 결과이다.

```
plain text: 0123456789ABCDEFFEDCBA98765432100A1B2C3D4E5F6
encryption


0A1B2C3D4E5F6DC1

Plain Text : [BD1BC8C700000000]
   Key     : [85E813540F0AB405]
round     left          right         key
    1  [0D01070A]  [55FD22A8]  [C842107B226C]
    2  [55FD22A8]  [DE4EAE7A]  [35602242C637]
    3  [DE4EAE7A]  [710B5CFB]  [A205801F2D8C]
    4  [710B5CFB]  [8D71848C]  [580215A851D1]
    5  [8D71848C]  [F9672AE1]  [05901843E227]
    6  [F9672AE1]  [EA76F8D8]  [0600E6F60D88]
    7  [EA76F8D8]  [5BD15CBB]  [BA480088135F]
    8  [5BD15CBB]  [B3110961]  [08232857F2A0]
    9  [B3110961]  [54D330A2]  [23A0C408EA4F]
   10  [54D330A2]  [3BA60F3E]  [18448276D494]
   11  [3BA60F3E]  [C999BDB5]  [7001388905EB]
   12  [C999BDB5]  [15CFFED8]  [8480058EFA01]
   13  [15CFFED8]  [C0C169B0]  [030A16724774]
   14  [C0C169B0]  [2F882850]  [2C30A099898A]
   15  [2F882850]  [5D7EB29B]  [920468C47611]
   16  [BB251D97]  [5D7EB29B]  [0088D3B998C1]
Cipher Text : [D76BB5E6EF78A04B]
Plain Text : [BD1BC8C700000001]
   Key     : [85E813540F0AB405]
round     left          right         key
    1  [0D01070A]  [55FD2228]  [C842107B226C]
    2  [55FD2228]  [CC6E8E7B]  [35602242C637]
    3  [CC6E8E7B]  [700150CD]  [A205801F2D8C]
    4  [700150CD]  [CD4990B5]  [580215A851D1]
    5  [CD4990B5]  [EA4A071B]  [05901843E227]
    6  [EA4A071B]  [637738A3]  [0600E6F60D88]
    7  [637738A3]  [2E63B3C3]  [BA480088135F]
    8  [2E63B3C3]  [304F37EE]  [08232857F2A0]
    9  [304F37EE]  [08E3ABAE]  [23A0C408EA4F]
   10  [08E3ABAE]  [A50CD707]  [18448276D494]
   11  [A50CD707]  [CE1E6E05]  [7001388905EB]
   12  [CE1E6E05]  [F83F4F68]  [8480058EFA01]
   13  [F83F4F68]  [92951A78]  [030A16724774]
   14  [92951A78]  [AEF717C2]  [2C30A099898A]
   15  [AEF717C2]  [50FF936E]  [920468C47611]
   16  [7522EC4D]  [50FF936E]  [0088D3B998C1]
Cipher Text : [693A6727E876E72C]
Plain Text : [BD1BC8C700000002]
   Key     : [85E813540F0AB405]
round     left          right         key
    1  [0D01078A]  [55CD02A9]  [C842107B226C]
    2  [55CD02A9]  [F20F343E]  [35602242C637]
    3  [F20F343E]  [FEA6B946]  [A205801F2D8C]
    4  [FEA6B946]  [18F39242]  [580215A851D1]
    5  [18F39242]  [BEDE008E]  [05901843E227]
    6  [BEDE008E]  [7C7D432F]  [0600E6F60D88]
    7  [7C7D432F]  [BBAF481B]  [BA480088135F]
    8  [BBAF481B]  [200F1879]  [08232857F2A0]
    9  [200F1879]  [D89D1AF2]  [23A0C408EA4F]
   10  [D89D1AF2]  [FD2843C2]  [18448276D494]
   11  [FD2843C2]  [AF2BCCB8]  [7001388905EB]
   12  [AF2BCCB8]  [E9D102D1]  [8480058EFA01]
   13  [E9D102D1]  [217043B0]  [030A16724774]
   14  [217043B0]  [369156F7]  [2C30A099898A]
   15  [369156F7]  [1C2BDFB5]  [920468C47611]
   16  [1947A72F]  [1C2BDFB5]  [0088D3B998C1]
Cipher Text : [7F3D9FE9CA27180E]
D648F08166D36DA4 97E6DDBF9E22D53C 7526B3D4847875CF
```

```
decryption
Plain Text : [BD1BC8C700000000]
   Key     : [85E813540F0AB405]
round      left          right         key
    1  [0D01070A]  [55FD22A8]  [C842107B226C]
    2  [55FD22A8]  [DE4EAE7A]  [35602242C637]
    3  [DE4EAE7A]  [710B5CFB]  [A205801F2D8C]
    4  [710B5CFB]  [8D71848C]  [580215A851D1]
    5  [8D71848C]  [F9672AE1]  [05901843E227]
    6  [F9672AE1]  [EA76F8D8]  [0600E6F60D88]
    7  [EA76F8D8]  [5BD15CBB]  [BA480088135F]
    8  [5BD15CBB]  [B3110961]  [08232857F2A0]
    9  [B3110961]  [54D330A2]  [23A0C408EA4F]
   10  [54D330A2]  [3BA60F3E]  [18448276D494]
   11  [3BA60F3E]  [C999BDB5]  [7001388905EB]
   12  [C999BDB5]  [15CFFED8]  [8480058EFA01]
   13  [15CFFED8]  [C0C169B0]  [030A16724774]
   14  [C0C169B0]  [2F882850]  [2C30A099898A]
   15  [2F882850]  [5D7EB29B]  [920468C47611]
   16  [BB251D97]  [5D7EB29B]  [0088D3B998C1]
Cipher Text : [D76BB5E6EF78A04B]
Plain Text : [BD1BC8C700000001]
   Key     : [85E813540F0AB405]
round      left          right         key
    1  [0D01070A]  [55FD2228]  [C842107B226C]
    2  [55FD2228]  [CC6E8E7B]  [35602242C637]
    3  [CC6E8E7B]  [700150CD]  [A205801F2D8C]
    4  [700150CD]  [CD4990B5]  [580215A851D1]
    5  [CD4990B5]  [EA4A071B]  [05901843E227]
    6  [EA4A071B]  [637738A3]  [0600E6F60D88]
    7  [637738A3]  [2E63B3C3]  [BA480088135F]
    8  [2E63B3C3]  [304F37EE]  [08232857F2A0]
    9  [304F37EE]  [08E3ABAE]  [23A0C408EA4F]
   10  [08E3ABAE]  [A50CD707]  [18448276D494]
   11  [A50CD707]  [CE1E6E05]  [7001388905EB]
   12  [CE1E6E05]  [F83F4F68]  [8480058EFA01]
   13  [F83F4F68]  [92951A78]  [030A16724774]
   14  [92951A78]  [AEF717C2]  [2C30A099898A]
   15  [AEF717C2]  [50FF936E]  [920468C47611]
   16  [7522EC4D]  [50FF936E]  [0088D3B998C1]
Cipher Text : [693A6727E876E72C]
Plain Text : [BD1BC8C700000002]
   Key     : [85E813540F0AB405]
round      left          right         key
    1  [0D01078A]  [55CD02A9]  [C842107B226C]
    2  [55CD02A9]  [F20F343E]  [35602242C637]
    3  [F20F343E]  [FEA6B946]  [A205801F2D8C]
    4  [FEA6B946]  [18F39242]  [580215A851D1]
    5  [18F39242]  [BEDE008E]  [05901843E227]
    6  [BEDE008E]  [7C7D432F]  [0600E6F60D88]
    7  [7C7D432F]  [BBAF481B]  [BA480088135F]
    8  [BBAF481B]  [200F1879]  [08232857F2A0]
    9  [200F1879]  [D89D1AF2]  [23A0C408EA4F]
   10  [D89D1AF2]  [FD2843C2]  [18448276D494]
   11  [FD2843C2]  [AF2BCCB8]  [7001388905EB]
   12  [AF2BCCB8]  [E9D102D1]  [8480058EFA01]
   13  [E9D102D1]  [217043B0]  [030A16724774]
   14  [217043B0]  [369156F7]  [2C30A099898A]
   15  [369156F7]  [1C2BDFB5]  [920468C47611]
   16  [1947A72F]  [1C2BDFB5]  [0088D3B998C1]
Cipher Text : [7F3D9FE9CA27180E]
0123456789ABCDEF FEDCBA9876543210 0A1B2C3D4E5F6
```

CTR 의 암호화 복호화 결과이다.

4. 성능분석 및 차이점

```
CBC encryption
CBC encryption elapse time : 30.823
CBC decryption
CBC decryption elapse time : 28.637
CTR encryption
CTR encryption elapse time : 29.8
CTR decryption
CTR decryption elapse time : 28.453
```

10mb

```
CBC encryption
CBC encryption elapse time : 58.789
CBC decryption
CBC decryption elapse time : 59.777
CTR encryption
CTR encryption elapse time : 60.66
CTR decryption
CTR decryption elapse time : 58.033

S:₩repos₩3-2₩DES₩x64₩Release₩DES.exe(프로
이 창을 닫으려면 아무 키나 누르세요...
```

20mb

```
CBC encryption elapse time : 85.769
CBC decryption elapse time : 82.52
CTR encryption elapse time : 84.642
CTR decryption elapse time : 86.068
```

30mb

```
CBC encryption elapse time : 116.761
CBC decryption elapse time : 115.163
CTR encryption elapse time : 110.495
CTR decryption elapse time : 107.957
```

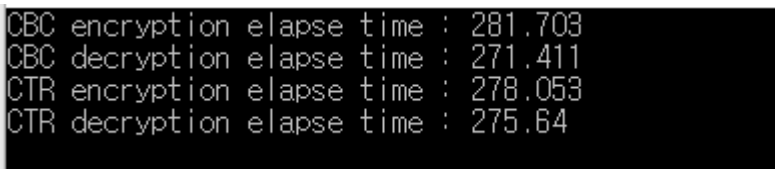
40mb

```
CBC encryption elapse time : 149.207
CBC decryption elapse time : 141.226
CTR encryption elapse time : 139.79
CTR decryption elapse time : 137.673
```

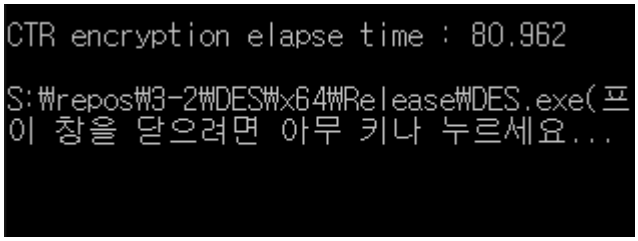
50mb

```
CBC encryption elapse time : 168.859
CBC decryption elapse time : 164.553
CTR encryption elapse time : 169.564
CTR decryption elapse time : 167.323
```

60mb



100mb



100mb 를 병렬처리하여 암호화.

CTR 과 CBC 둘 다 암호화와 복호화를 했을 때 엄청 유의미할 정도로 크게 성능차이가 나지는 않은 것 같다. 약간씩 오차가 있지만 전체적으로 CTR 이 조금 미세하게 빠른 것 같다.
CTR 의 암호화를 병렬처리하여 실행하였을 경우 눈에 띄게 속도가 줄어듦을 알 수 있다. 해당프로그램에서 4 개의 스레드로 멀티스레딩을 하여 처리하였는데 3~4 배정도 차이가 남을 알 수 있다.

CBC 와 CTR 은 둘 다 블록 암호이다. CBC 는 이전 블록의 암호문과 XOR 하는 과정을 거치고 CTR 은 counter 를 이용하여 XOR 연산을 진행한다. CTR 이 CBC 와 가지는 가장 큰 차이점은 바로 counter 를 이용하는 데에 있다. 우선 CTR 은 블록 암호를 스트림 암호처럼 바꾼다. 따라서 암호화와 복호화 과정이 완전히 동일하기 때문에 프로그램 구현이 매우 쉽다. 또한 counter 가 블록 개수와 동일하고 순서도 동일하기 때문에 CBC 처럼 한번에 연달아 암호화 복호화 하지 않고, 원하는 블록만 별도로 암호화 복호화 할 수 있다. 또한 그래서 CTR 의 경우 병렬처리가 가능하다. 병렬처리를 통해서 암호화/복호화 속도를 매우 빠르게 끌어 올릴 수 있을 것이다.

5. 소감

블록을 암호화 하는 방식은 이전에 만들어 놓은 des 를 사용하여 어렵지 않았다. 이번 과제를 하면서 제일 까다로웠던 것은 CBC 암호의 복호화 부분이었던 것 같다. 처음에 완전히 암호화의 역순으로 생각했다가 한참을 왜 안되지... 했다. 역시 개념이 중요 한 것 같다. 생각 이상으로 암호화 하는데 시간이 오래 걸린다는 것이 확 체감이 되었다. 100mb 크기의 파일을 처리하는데에도 5 분가량의 시간이 걸리는데 만약 내 하드디스크의 자료를 모두 암호화 한다 생각하면 어질어질 해 진다.

병렬처리를 어떻게 어떻게 하기는 하였으나, 애초에 암호화 복호화 과정을 연이어 하는 것으로 생각하고 클래스 설계를 하여 암호화 만 병렬처리로 테스트 해 볼 수 있었다. 클래스를 변경하면 되었지만 그거까지 할 시간이 없어서 어쩔 수 없이 암호화만 병렬처리를 하는것에서 멈춰야했다. 암호화만 테스트 해 보았지만, 4 개의 스레드로 수행했을 때 시간이 많이 줄어드는 것으로 보아 복호화도 마찬가지 일 것으로 보인다.용량이 커질수록 병렬처리가 가능 한 CTR 이 CBC 에 비해 많이 유리하리라는 생각이 아주 많이 들었다.