

# AES

# CBC ECB CTR OCF CFB

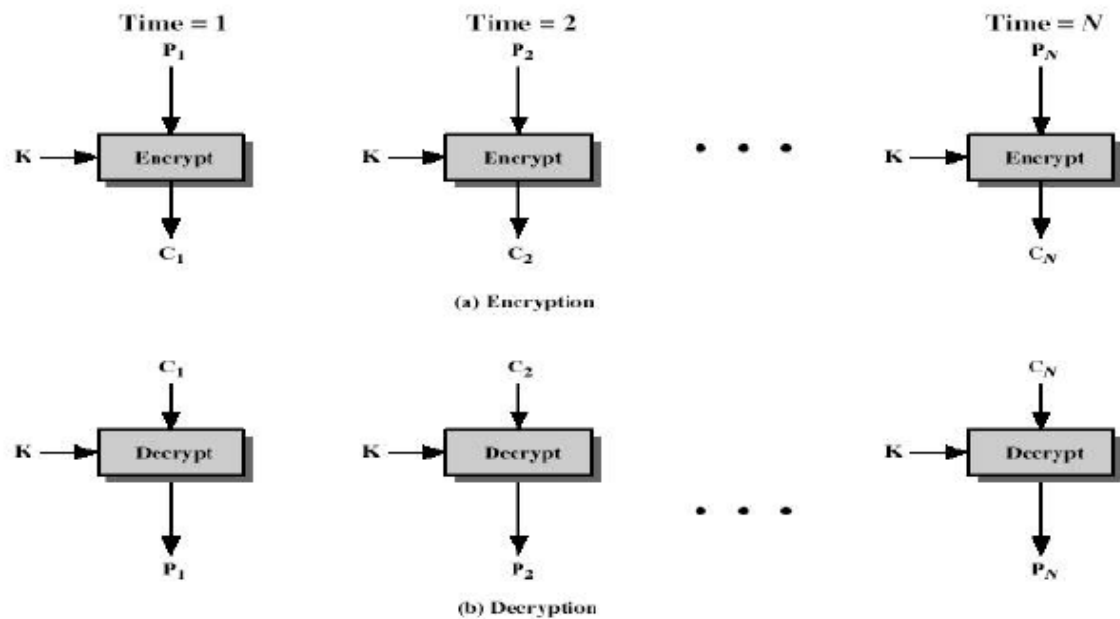
“ AES

CBC ECB CTR OCF CFB

1. Electronic Codebook Book (ECB)
2. Cipher Block Chaining (CBC)
3. Counter (CTR)
4. Cipher FeedBack (CFB)
5. Output FeedBack (OFB)

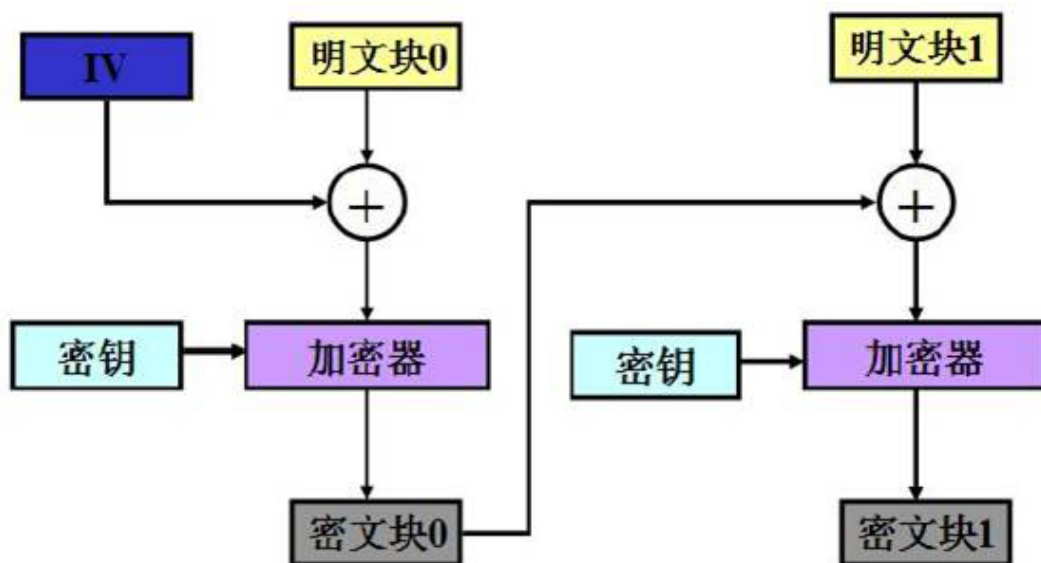
## 1. Electronic Codebook Book (ECB)

### ECB



## 2. Cipher Block Chaining (CBC)

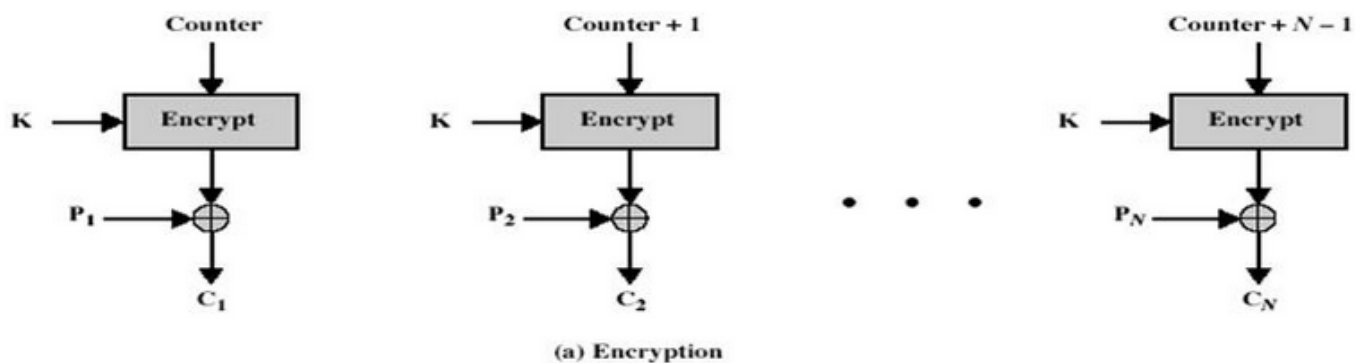
# CBC



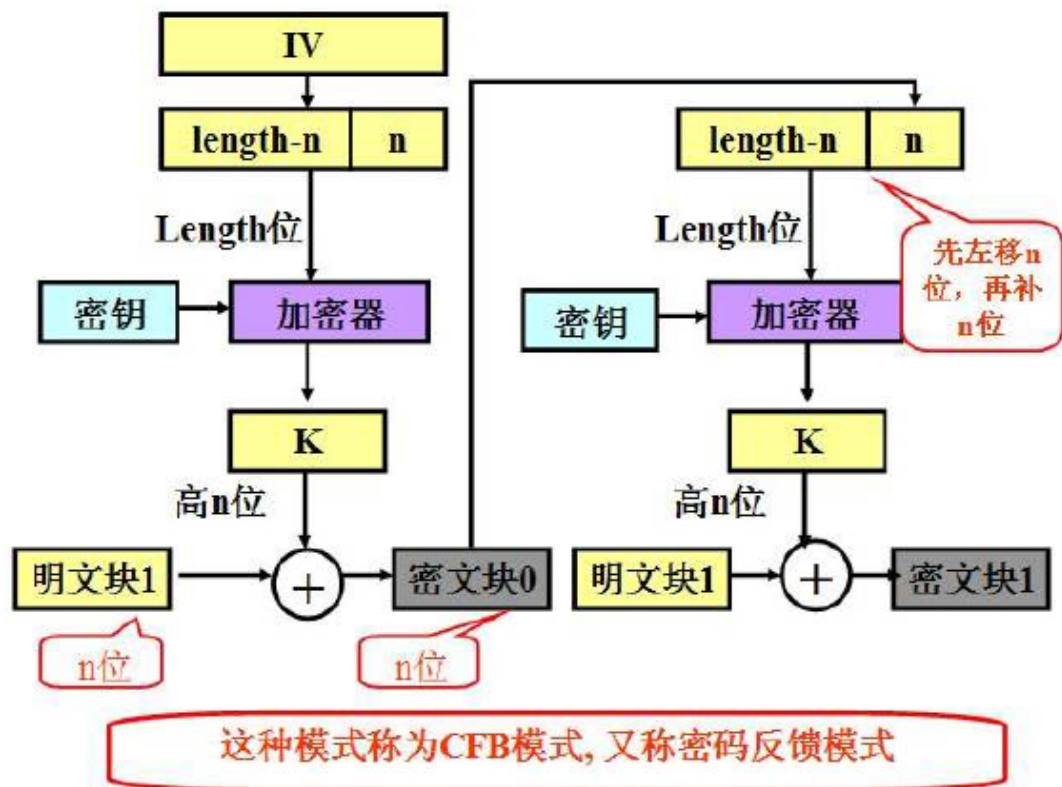
这种模式称为CBC模式, 又密码分组链接

## 3. Counter (CTR)

CTR

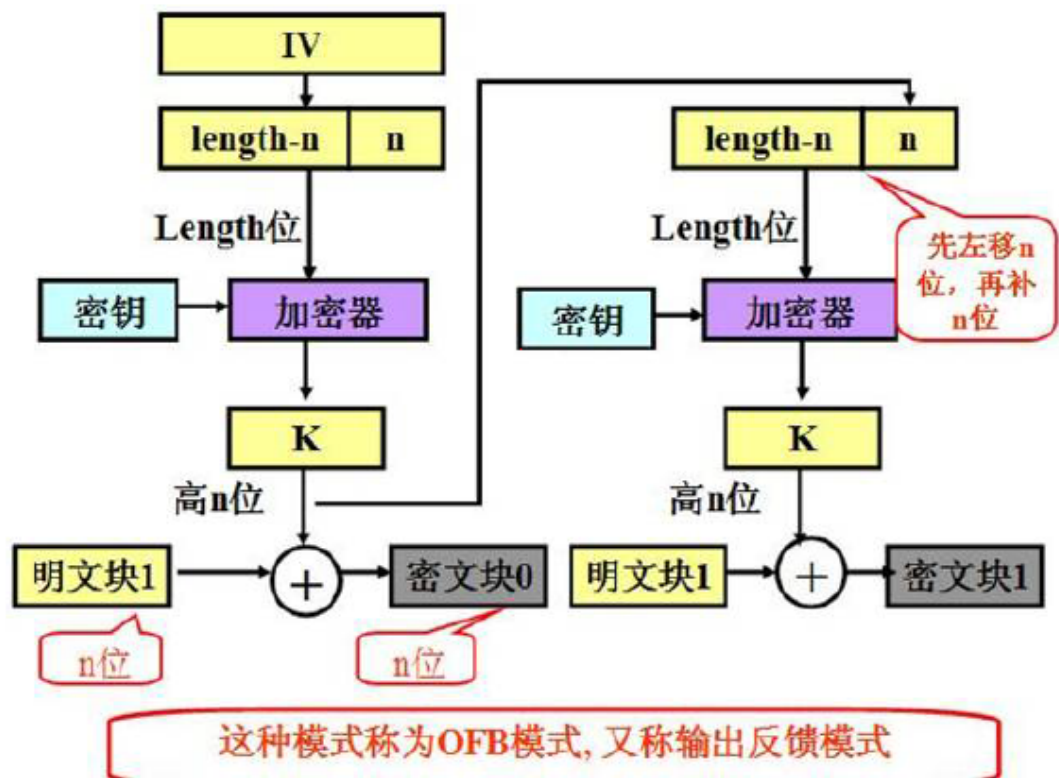


## 4. Cipher FeedBack (CFB)



## 5. Output FeedBack (OFB)

# OFB



C++

```
/**
 * @author stardust
 * @time 2013-10-10
 * @param AES
 */
#include <iostream>
using namespace std;

// , 16 1 0
int dataLen = 16; //
int encLen = 4; //
int encTable[4] = {1, 0, 1, 0}; //
int data[16] = {1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0}; //
int ciphertext[16]; //

//
void encode(int arr[])
{

```

```

    for(int i=0; i<encLen; i++)
    {
        arr[i] = arr[i] ^ encTable[i];
    }
}

//      4
void ECB(int arr[])
{
    //
    int a[4][4];
    int dataCount = 0; //
    for(int k=0; k<4; k++)
    {
        for(int t=0; t<4; t++)
        {
            a[k][t] = data[dataCount];
            dataCount++;
        }
    }
    dataCount = 0; //
    for(int i=0; i<dataLen; i=i+encLen)
    {
        int r = i/encLen; //
        int l = 0; //
        int encQue[4]; //
        for(int j=0; j<encLen; j++)
        {
            encQue[j] = a[r][l];
            l++;
        }
        encode(encQue); //
        //
        for(int p=0; p<encLen; p++)
        {
            ciphertext[dataCount] = encQue[p];
            dataCount++;
        }
    }
    cout<<"ECB      "<<endl;
    for(int t1=0; t1<dataLen; t1++) //

```

```

{
    if(t1!=0 && t1%4==0)
        cout<<endl;
    cout<<ciiphertext[ t1] <<" ";
}
cout<<endl;
cout<<"-----" <<endl;
}

//CBC
//      4
void CCB(int arr[])
{
    //
    int a[4][4];
    int dataCount = 0;    //
    for( int k=0; k<4; k++)
    {
        for( int t=0; t<4; t++)
        {
            a[k][t] = data[ dataCount];
            dataCount++;
        }
    }
    dataCount = 0; //

    int init[4] = {1,1,0,0};    //
    //
    for( int i=0; i<dataLen; i=i+encLen)
    {
        int r = i/encLen; //
        int l = 0; //
        int encQue[4]; //
        //
        for( int k=0; k<encLen; k++)
        {
            a[r][k] = a[r][k] ^ init[k];
        }
        // Key
        for( int j=0; j<encLen; j++)
        {

```

```

        encQue[ j] = a[r][ j];
    }
    encode( encQue); //
    //
    for( int p=0; p<encLen; p++)
    {
        ciphertext[ dataCount] = encQue[ p];
        dataCount++;
    }
    //
    for( int t=0; t<encLen; t++)
    {
        init[ t] = encQue[ t];
    }
}

cout<<"CCB      " <<endl;
for( int t1=0; t1<dataLen; t1++) //
{
    if( t1!=0 && t1%4==0)
        cout<<endl;
    cout<<ciphertext[ t1] <<" ";
}
cout<<endl;
cout<<"-----" <<endl;
}

//CTR
//      4
void CTR( int arr[])
{
    //
    int a[ 4][ 4];
    int dataCount = 0; //
    for( int k=0; k<4; k++)
    {
        for( int t=0; t<4; t++)
        {
            a[ k][ t] = data[ dataCount];
            dataCount++;

```

```

    }
}
dataCount = 0; //

int init[4][4] = {{1, 0, 0, 0}, {0, 0, 0, 1}, {0, 0, 1, 0}, {0, 1, 0, 0}}; //
int l = 0; //
//
for( int i=0; i<dataLen; i=i+encLen)
{
    int r = i/encLen; //
    int encQue[4]; //
    //
    for( int t=0; t<encLen; t++)
    {
        encQue[t] = init[r][t];
    }
    encode( encQue); // key
    //
    for( int k=0; k<encLen; k++)
    {
        encQue[k] = encQue[k] ^ a[l][k];
    }
    l++;

    //
    for( int p=0; p<encLen; p++)
    {
        ciphertext[ dataCount] = encQue[ p];
        dataCount++;
    }
}

cout<<"CTR      " <<endl;
for( int t1=0; t1<dataLen; t1++) //
{
    if( t1!=0 && t1%4==0)
        cout<<endl;
    cout<<ciphertext[ t1] <<" ";
}
cout<<endl;

```



```

        cout<<"-----" <<endl;
    }

//CFB
//      4
void CFB(int arr[])
{
    //      , 2 * 8
    int a[8][2];
    int dataCount = 0; //
    for(int k=0; k<8; k++)
    {
        for(int t=0; t<2; t++)
        {
            a[k][t] = data[dataCount];
            dataCount++;
        }
    }
    dataCount = 0; //
    int lv[4] = {1, 0, 1, 1}; //
    int encQue[2]; //K
    int k[4]; //K

    for(int i=0; i<2 * encLen; i++) //
    {
        // K
        for(int vk=0; vk<encLen; vk++)
        {
            k[vk] = lv[vk];
        }
        encode(k);
        for(int k2=0; k2<2; k2++)
        {
            encQue[k2] = k[k2];
        }
        //K
        for(int j=0; j<2; j++)
        {
            ciphertext[dataCount] = a[dataCount/2][j] ^ encQue[j];
            dataCount++;
        }
    }
}

```

```

        //lv
        lv[0] = lv[2];
        lv[1] = lv[3];
        lv[2] = ciphertext[dataCount-2];
        lv[3] = ciphertext[dataCount-1];
    }

    cout<<" CFB      "<<endl;
    for(int t1=0; t1<dataLen; t1++) //
    {
        if(t1!=0 && t1%4==0)
            cout<<endl;
        cout<<ciphertext[t1]<<" ";
    }
    cout<<endl;
    cout<<"-----"<<endl;
}

//OFB
//      4
void OFB(int arr[])
{
    //      , 2 * 8
    int a[8][2];
    int dataCount = 0; //
    for(int k=0; k<8; k++)
    {
        for(int t=0; t<2; t++)
        {
            a[k][t] = data[dataCount];
            dataCount++;
        }
    }

    dataCount = 0; //
    int lv[4] = {1,0,1,1}; //
    int encQue[2]; //K
    int k[4]; //K

    for(int i=0; i<2 * encLen; i++) //
    {
        // K

```

```

    for( int vk=0; vk<encLen; vk++)
    {
        k[ vk]  = lv[ vk];
    }
    encode( k);
    for( int k2=0; k2<2; k2++)
    {
        encQue[ k2]  = k[ k2];
    }
    //K
    for( int j=0; j<2; j++)
    {
        ciphertext[ dataCount]  = a[ dataCount/2][ j]  ^ encQue[ j];
        dataCount++;
    }
    //lv
    lv[ 0]  = lv[ 2];
    lv[ 1]  = lv[ 3];
    lv[ 2]  = encQue[ 0];
    lv[ 3]  = encQue[ 1];
}

cout<<" CFB          " <<endl;
for( int t1=0; t1<dataLen; t1++) //
{
    if( t1!=0 && t1%4==0)
        cout<<endl;
    cout<<ciphertext[ t1] <<" ";
}
cout<<endl;
cout<<"-----" <<endl;
}

void printData()
{
    cout<<"    AES          " <<endl;
    cout<<"-----" <<endl;
    cout<<"    " <<endl;
    for( int t1=0; t1<dataLen; t1++) //
    {

```

```
        if(t1!=0 && t1%4==0)
            cout<<endl;
        cout<<data[t1]<<" ";
    }
    cout<<endl;
    cout<<"-----"<<endl;
}

int main()
{
    printData();
    ECB(data);
    CCB(data);
    CTR(data);
    CFB(data);
    OFB(data);
    return 0;
}
```

---

Revision #2

Created 26 July 2021 07:20:12 by

Updated 26 July 2021 07:23:35 by