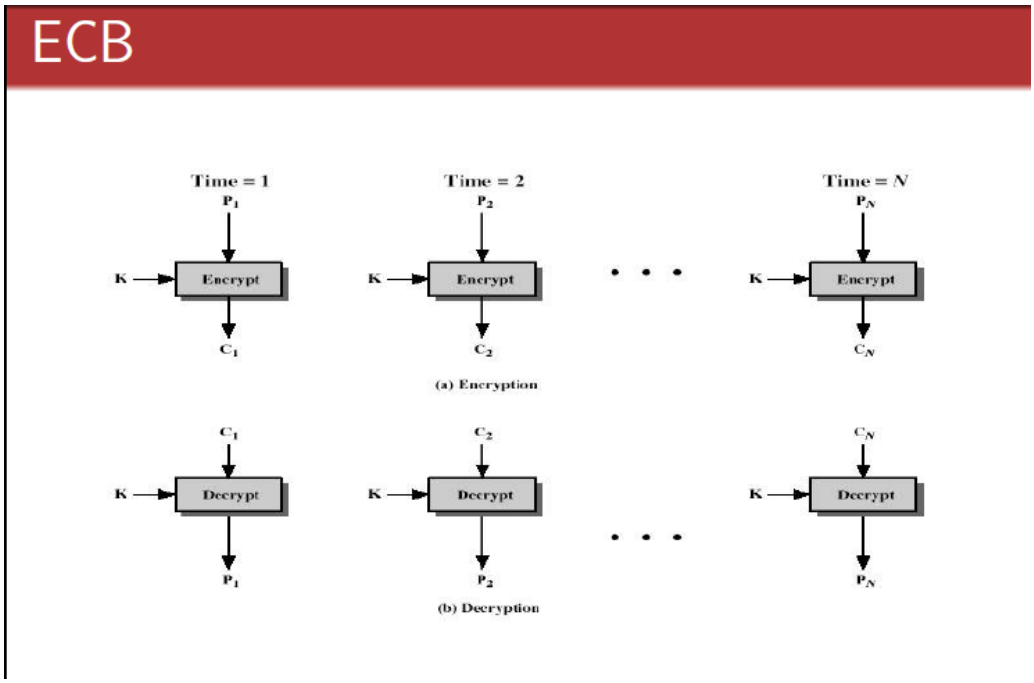
AES五种加密模式（CBC、ECB、CTR、OCF、CFB）

分组密码有五种工作体制：1.电码本模式（Electronic Codebook Book (ECB)）；2.密码分组链接模式（Cipher Block Chaining (CBC)）；3.计算器模式（Counter (CTR)）；4.密码反馈模式（Cipher FeedBack (CFB)）；5.输出反馈模式（Output FeedBack (OFB)）。
以下逐一介绍一下：

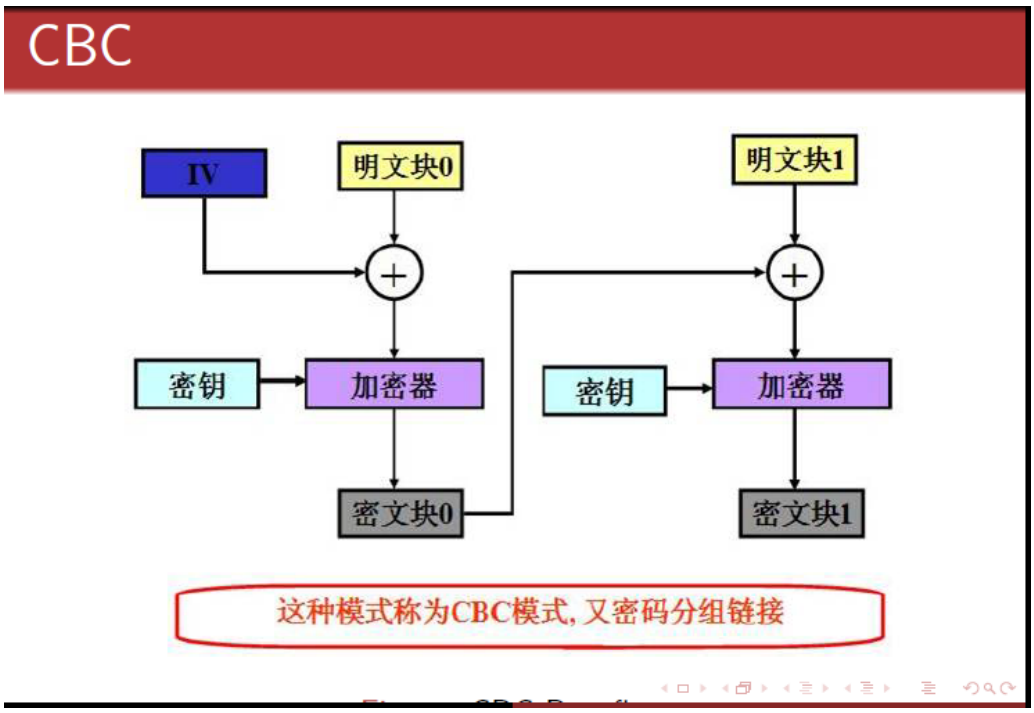
1.电码本模式（Electronic Codebook Book (ECB)

这种模式是将整个明文分成若干段相同的小段，然后对每一小段进行加密。



2.密码分组链接模式（Cipher Block Chaining (CBC)）

这种模式是先将明文切分成若干小段，然后每一小段与初始块或者上一段的密文段进行异或运算后，再与密钥进行加密。



3.计算器模式（Counter (CTR)）

计算器模式不常见，在CTR模式中， 有一个自增的算子，这个算子用密钥加密之后的输出和明文异或的结果得到密文，相当于一次一密。这种加密方式简单快速，安全可靠，而且可以并行加密，但是在计算器不能维持很长的情况下，密钥只能使用一次。CTR的示意图如下所示：

昵称：月之星狼
园龄：7年6个月
粉丝：11
关注：6
+加关注

< 2020年6月 >
日 一 二 三 四 五 六
31 1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 1 2 3 4
5 6 7 8 9 10 11

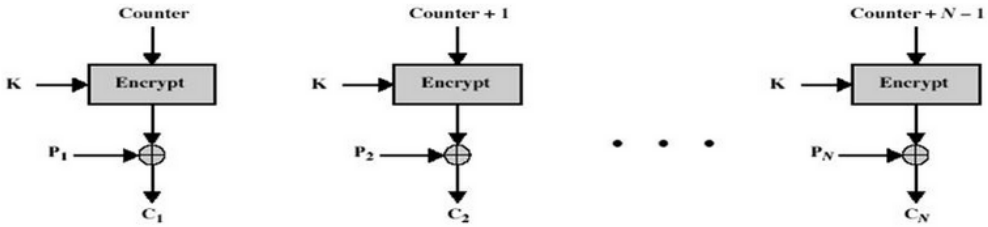
搜索
找找看
谷歌搜索

常用链接
我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签
iOS(4)
Android(4)
ARC机制(1)
ARC使用(1)
BroadcastReceiver(1)
delegate(1)
demo(1)
handler(1)
HTML(1)
2014(1)
更多

随笔分类
Android(6)
IOS(6)
JQuery(2)
每天有感(14)

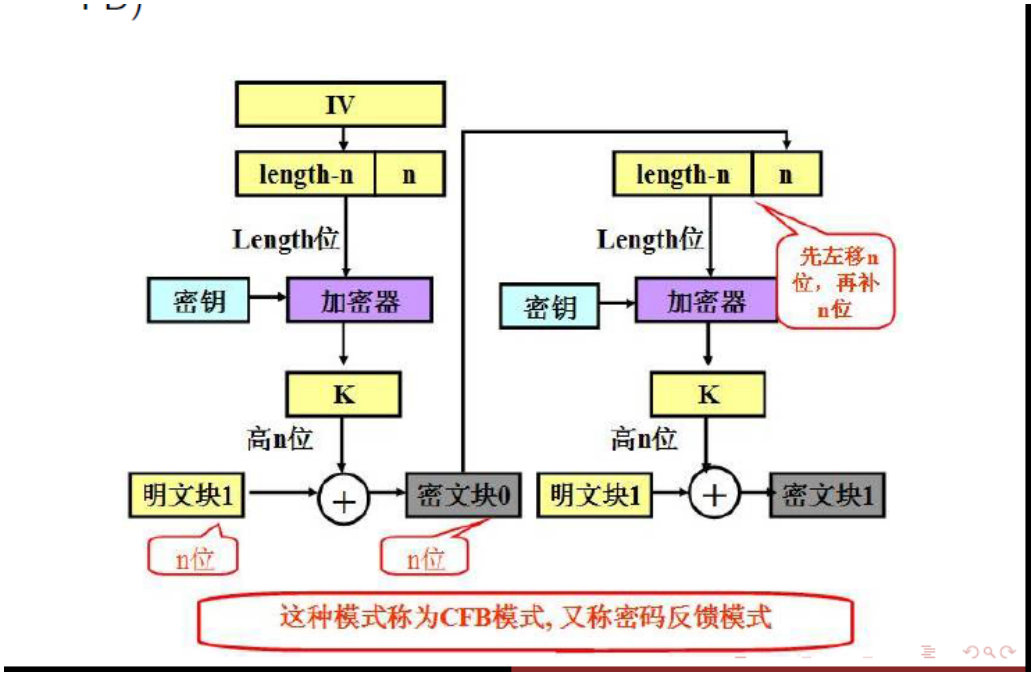
随笔档案
2015年1月(4)
2014年11月(2)
2014年8月(2)
2014年5月(7)



(a) Encryption

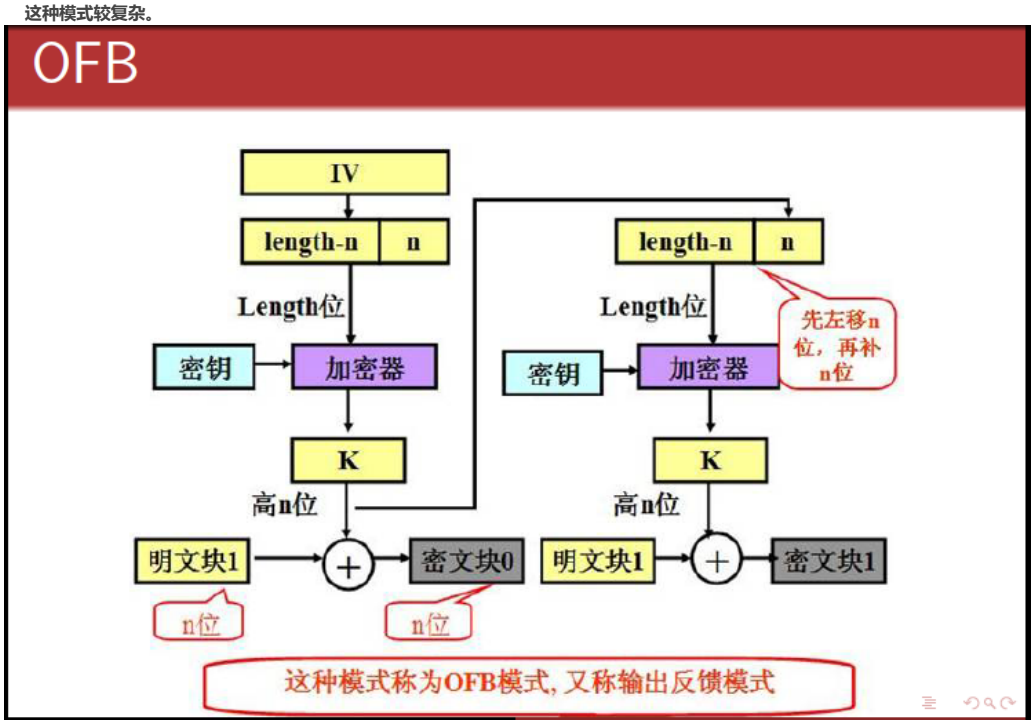
4.密码反馈模式（Cipher FeedBack (CFB)）

这种模式较复杂。



5.输出反馈模式（Output FeedBack (OFB)）

这种模式较复杂。



以下附上C++源代码：

```
/**
*@autho stardust
```

2014年3月(1)
2013年10月(1)
2013年9月(4)
2013年8月(5)
2013年7月(3)
2013年6月(1)
2013年5月(2)
2013年4月(1)
2013年1月(2)

最新评论

1. Re:AES五种加密模式（CBC、ECB、CTR、OCF、CFB）
没有AES-GCM模式
--mul
2. Re:java自己写的简单聊天工具SimpleQQ感悟
能不能分享一下exe文件,谢谢
--Minion2005
3. Re:AES五种加密模式（CBC、ECB、CTR、OCF、CFB）
很强大
--细品人生
4. Re:iOS开发系列-ARC浅解
@ tiger_zh可以坚持多久就坚持多久，大家共勉吧...
--月之星狼
5. Re:iOS开发系列-ARC浅解
看看博主能坚持到什么时候，与君共勉。
--tiger_zh

阅读排行榜

1. AES五种加密模式（CBC、ECB、CTR、OCF、CFB）(107992)
2. java自己写的简单聊天工具SimpleQQ感悟(5147)
3. 2014阿里实习招聘笔试有感(2517)
4. Android复杂自定义Listview实现(2403)
5. Android SQLite最简单demo实现（增删查改）(1618)

评论排行榜

1. 2014阿里实习招聘笔试有感(9)
2. 大三，一点回忆，一点难忘(4)
3. AES五种加密模式（CBC、ECB、CTR、OCF、CFB）(3)
4. iOS开发系列-ARC浅解(2)
5. java自己写的简单聊天工具SimpleQQ感悟(1)

推荐排行榜

1. 2014阿里实习招聘笔试有感(3)

```
*@time 2013-10-10
*@param 实现AES五种加密模式的测试
*/
#include <iostream>
using namespace std;

//加密编码过程函数,16位1和0
int dataLen = 16;   //需要加密数据的长度
int encLen = 4;     //加密分段的长度
int encTable[4] = {1,0,1,0};  //置换表
int data[16] = {1,0,0,1,0,0,0,1,1,1,1,1,0,0,0,0}; //明文
int ciphertext[16]; //密文

//切片加密函数
void encode(int arr[])
{
    for(int i=0;i<encLen;i++)
    {
        arr[i] = arr[i] ^ encTable[i];
    }
}

//电码本模式加密，4位分段
void ECB(int arr[])
{
    //数据明文切片
    int a[4][4];
    int dataCount = 0;  //位置变量
    for(int k=0;k<4;k++)
    {
        for(int t=0;t<4;t++)
        {
            a[k][t] = data[dataCount];
            dataCount++;
        }
    }
    dataCount = 0;//重置位置变量
    for(int i=0;i<dataLen;i=i+encLen)
    {
        int r = i/encLen;//行
        int l = 0;//列
        int encQue[4]; //编码片段
        for(int j=0;j<encLen;j++)
        {
            encQue[j] = a[r][l];
            l++;
        }
        encode(encQue); //切片加密
        //添加到密文表中
        for(int p=0;p<encLen;p++)
        {
            ciphertext[dataCount] = encQue[p];
            dataCount++;
        }
    }
    cout<<"ECB加密的密文为："<<endl;
    for(int t1=0;t1<dataLen;t1++) //输出密文
    {
        if(t1!=0 && t1%4==0)
            cout<<endl;
        cout<<ciphertext[t1]<<" ";
    }
    cout<<endl;
    cout<<"-----------------------------------------------"<<endl;
}

//CBC
//密码分组链接模式，4位分段
void CCB(int arr[])
{
    //数据明文切片
    int a[4][4];
    int dataCount = 0;  //位置变量
    for(int k=0;k<4;k++)
    {
        for(int t=0;t<4;t++)
        {
            a[k][t] = data[dataCount];
            dataCount++;
        }
    }
```

2. AES五种加密模式（CBC、ECB、CTR、OCF、CFB）(3)
3. 大三，一点回忆，一点难忘(2)
4. Java EE学习路线(1)

```
    dataCount = 0;//重置位置变量

    int init[4] = {1,1,0,0};  //初始异或运算输入
    //初始异或运算
    for(int i=0;i<dataLen;i=i+encLen)
    {
        int r = i/encLen;//行
        int l = 0;//列
        int encQue[4]; //编码片段
        //初始化异或运算
        for(int k=0;k<encLen;k++)
        {
            a[r][k] = a[r][k] ^ init[k];
        }
         //与Key加密的单切片
        for(int j=0;j<encLen;j++)
        {
            encQue[j] = a[r][j];
        }
        encode(encQue); //切片加密
        //添加到密文表中
        for(int p=0;p<encLen;p++)
        {
            ciphertext[dataCount] = encQue[p];
            dataCount++;
        }
        //变换初始输入
        for(int t=0;t<encLen;t++)
        {
            init[t] = encQue[t];
        }
    }


    cout<<"CCB加密的密文为："<<endl;
    for(int t1=0;t1<dataLen;t1++) //输出密文
    {
        if(t1!=0 && t1%4==0)
            cout<<endl;
        cout<<ciphertext[t1]<<" ";
    }
    cout<<endl;
    cout<<"-----------------------------------------------"<<endl;
}

//CTR
//计算器模式，4位分段
void CTR(int arr[])
{
    //数据明文切片
    int a[4][4];
    int dataCount = 0;  //位置变量
    for(int k=0;k<4;k++)
    {
        for(int t=0;t<4;t++)
        {
            a[k][t] = data[dataCount];
            dataCount++;
        }
    }
    dataCount = 0;//重置位置变量

    int init[4][4] = {{1,0,0,0},{0,0,0,1},{0,0,1,0},{0,1,0,0}};  //算子表
    int l = 0; //明文切片表列
    //初始异或运算
    for(int i=0;i<dataLen;i=i+encLen)
    {
        int r = i/encLen;//行
        int encQue[4]; //编码片段
        //将算子切片
        for(int t=0;t<encLen;t++)
        {
            encQue[t] = init[r][t];
        }
        encode(encQue); //算子与key加密
        //最后的异或运算
        for(int k=0;k<encLen;k++)
        {
            encQue[k] = encQue[k] ^ a[l][k];
        }
        l++;
```

```
        //添加到密文表中
        for(int p=0;p<encLen;p++)
        {
            ciphertext[dataCount] = encQue[p];
            dataCount++;
        }
    }


    cout<<"CTR加密的密文为: "<<endl;
    for(int t1=0;t1<dataLen;t1++) //输出密文
    {
        if(t1!=0 && t1%4==0)
            cout<<endl;
        cout<<ciphertext[t1]<<" ";
    }
    cout<<endl;
    cout<<"-----------------------------------------------"<<endl;
}

//CFB
//密码反馈模式, 4位分段
void CFB(int arr[])
{
    //数据明文切片,切成2 * 8 片
    int a[8][2];
    int dataCount = 0;  //位置变量
    for(int k=0;k<8;k++)
    {
        for(int t=0;t<2;t++)
        {
            a[k][t] = data[dataCount];
            dataCount++;
        }
    }
    dataCount = 0;  //恢复初始化设置
    int lv[4] = {1,0,1,1};  //初始设置的位移变量
    int encQue[2]; //K的高两位
    int k[4]; //K

    for(int i=0;i<2 * encLen;i++) //外层加密循环
    {
        //产生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与数据明文异或产生密文
        for(int j=0;j<2;j++)
        {
            ciphertext[dataCount] = a[dataCount/2][j] ^ encQue[j];
            dataCount++;
        }
        //lv左移变换
        lv[0] = lv[2];
        lv[1] = lv[3];
        lv[2] = ciphertext[dataCount-2];
        lv[3] = ciphertext[dataCount-1];
    }

    cout<<"CFB加密的密文为: "<<endl;
    for(int t1=0;t1<dataLen;t1++) //输出密文
    {
        if(t1!=0 && t1%4==0)
            cout<<endl;
        cout<<ciphertext[t1]<<" ";
    }
    cout<<endl;
    cout<<"-----------------------------------------------"<<endl;
}

//OFB
//输出反馈模式, 4位分段
void OFB(int arr[])
{
```

```
    //数据明文切片,切成2 * 8 片
    int a[8][2];
    int dataCount = 0;  //位置变量
    for(int k=0;k<8;k++)
    {
        for(int t=0;t<2;t++)
        {
            a[k][t] = data[dataCount];
            dataCount++;
        }
    }
    dataCount = 0;  //恢复初始化设置
    int lv[4] = {1,0,1,1};  //初始设置的位移变量
    int encQue[2]; //K的高两位
    int k[4]; //K

    for(int i=0;i<2 * encLen;i++) //外层加密循环
    {
        //产生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与数据明文异或产生密文
        for(int j=0;j<2;j++)
        {
            ciphertext[dataCount] = a[dataCount/2][j] ^ encQue[j];
            dataCount++;
        }
        //lv左移变换
        lv[0] = lv[2];
        lv[1] = lv[3];
        lv[2] = encQue[0];
        lv[3] = encQue[1];
    }

    cout<<"CFB加密的密文为："<<endl;
    for(int t1=0;t1<dataLen;t1++) //输出密文
    {
        if(t1!=0 && t1%4==0)
            cout<<endl;
        cout<<ciphertext[t1]<<" ";
    }
    cout<<endl;
    cout<<"-----------------------------------------------"<<endl;
}


void printData()
{
    cout<<"以下示范AES五种加密模式的测试结果："<<endl;
    cout<<"-----------------------------------------------"<<endl;
    cout<<"明文为："<<endl;
    for(int t1=0;t1<dataLen;t1++) //输出密文
    {
        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;
}
```