AES五种加密模式（CBC、ECB、CTR、OCF、CFB）- 月之星狼

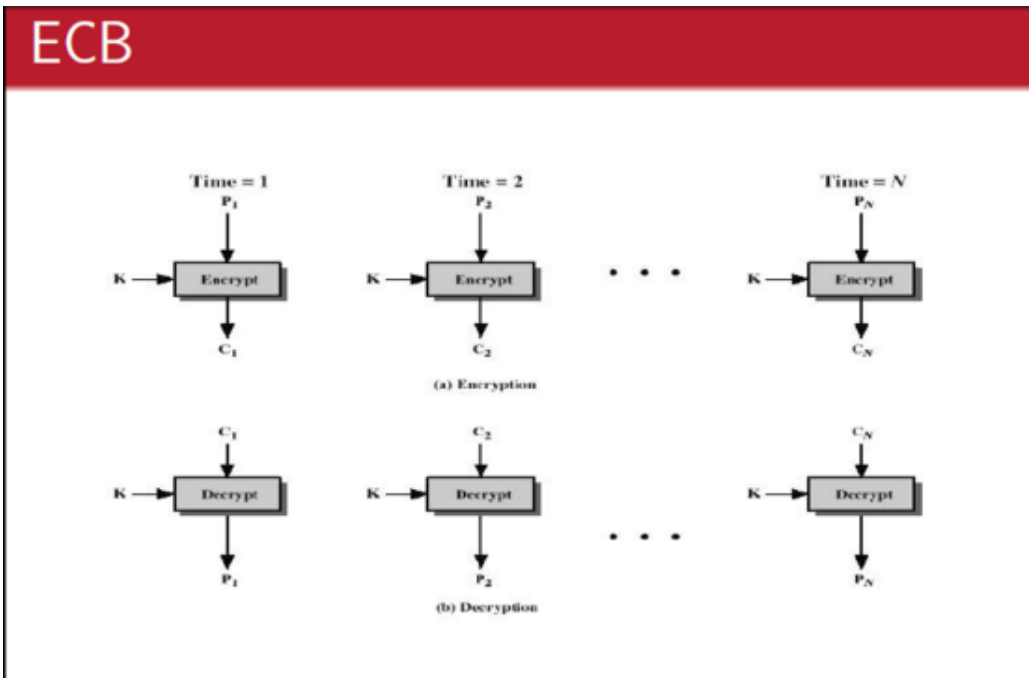
时间 2013-10-12 19:49:00 博客园-原创精华区 (/sites/Fn2umm)

原文
http://www.cnblogs.com/starwolf/p/3365834.html (http://www.cnblogs.com/starwolf/p/3365834.html?utm_source=tuicool&utm_medium=referral)
主题 加密解密 (/topics/11100078)
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：

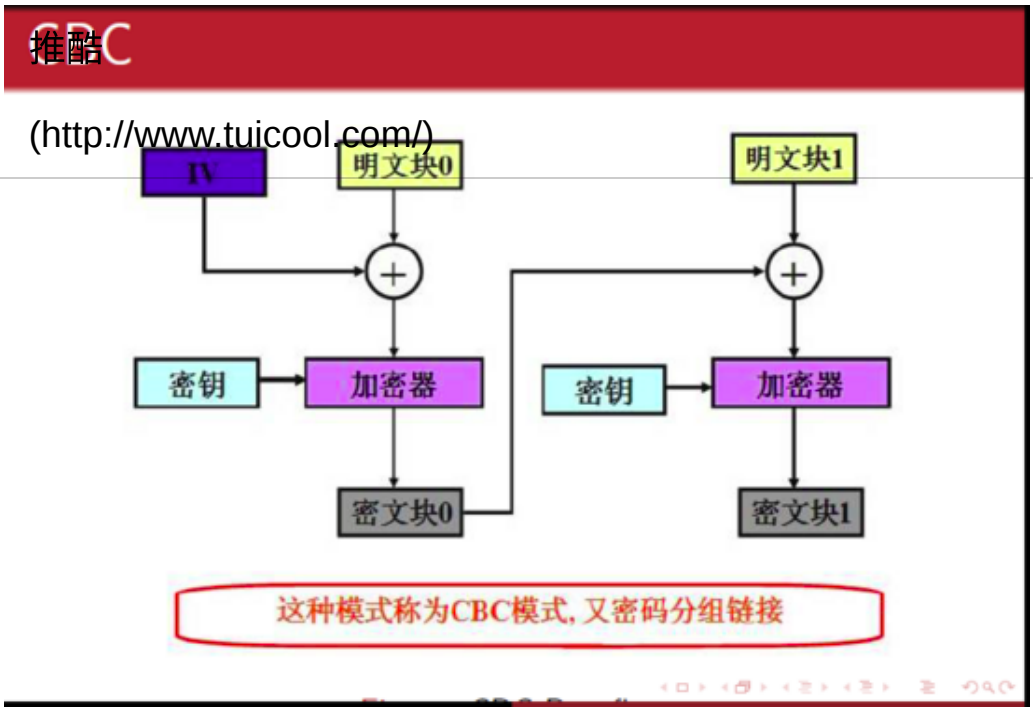
1.电码本模式（ Electronic Codebook Book (ECB)

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。



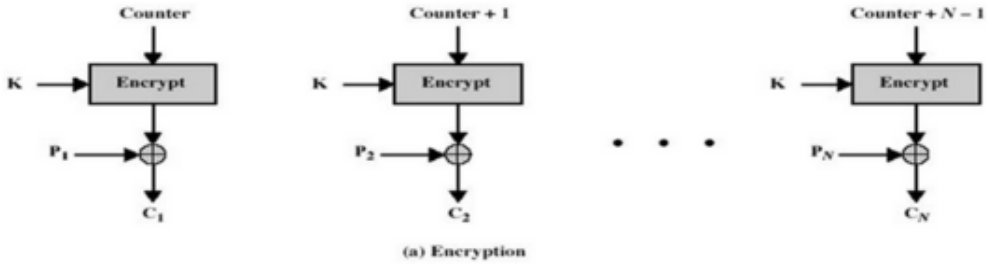
2.密码分组链接模式（Cipher Block Chaining (CBC)）

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。



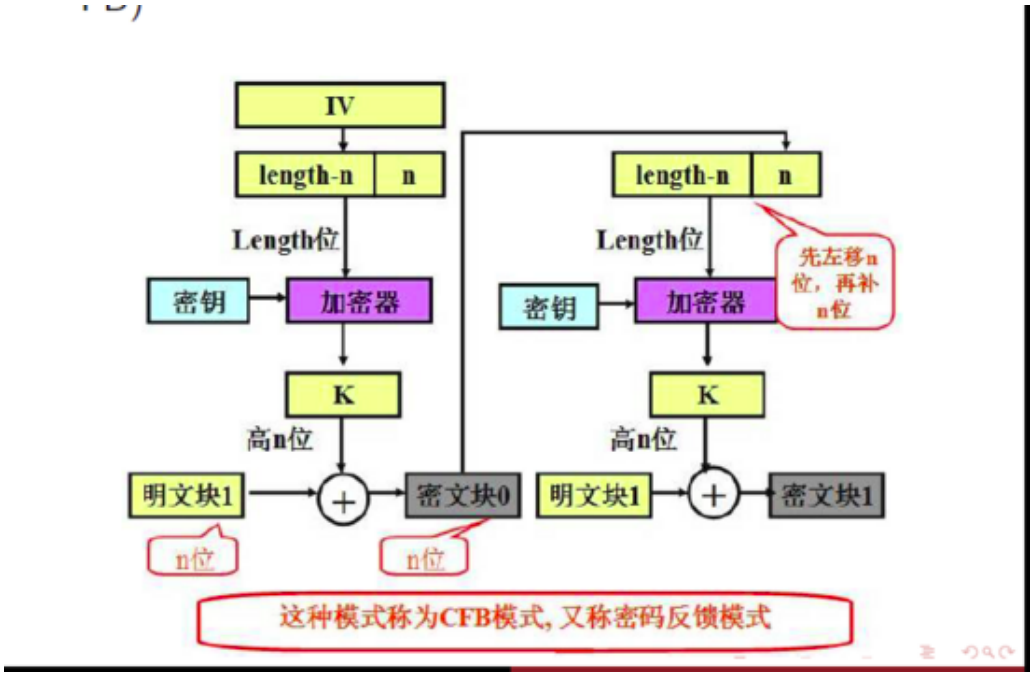
3.计算器模式（Counter (CTR)）

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：



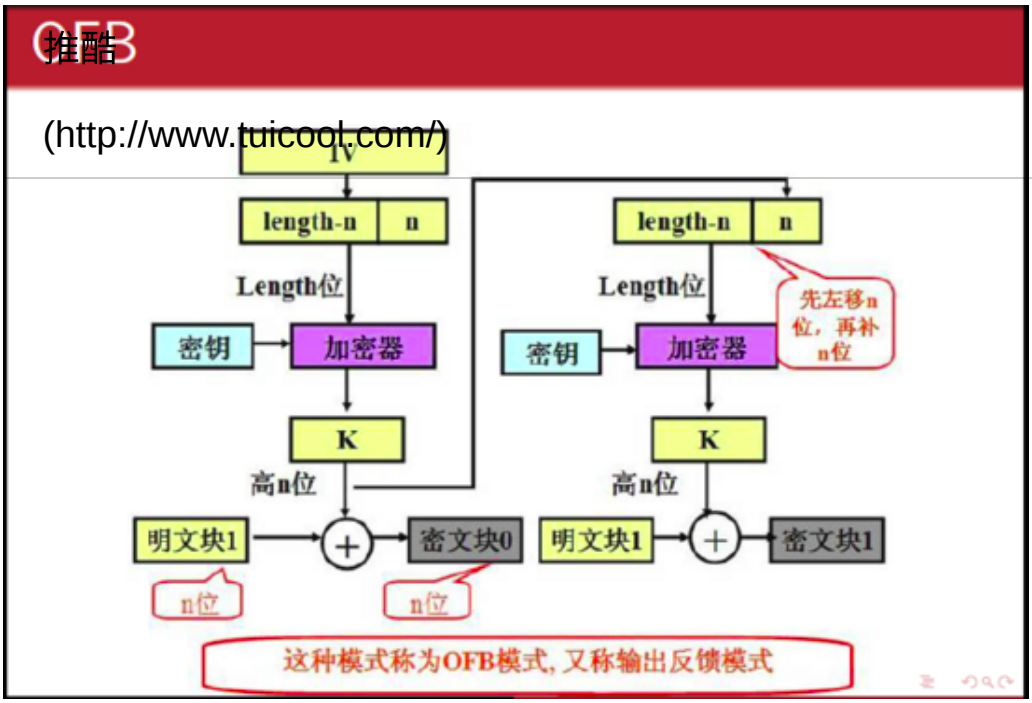
4.密码反馈模式（Cipher FeedBack (CFB)）

这种模式较复杂。



5.输出反馈模式（Output FeedBack (OFB)）

这种模式较复杂。


OFB
IV
length-n
n
Length位
密钥
加密器
K
高n位
明文块1
密文块0
n位
n位
length-n
n
Length位
密钥
加密器
先左移n位，再补n位
K
高n位
明文块1
密文块1
这种模式称为OFB模式，又称输出反馈模式

以下附上C++源代码：

```
/*
*@autho stardust
*@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++;
        }
    }
    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位移变换
        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;
}
```

推酷
(http://www.tuicool.com/)

推荐文章

- 1. 突破GIL..... - 转载 - Quicklib 源码分析 3、4 (/articles/32eeUnv)
- 2. C语言中数组与指针的关系 (/articles/j6f6BbV)
- 3. 突破GIL？！ - 转载 - Quicklib 源码分析 1 (/articles/ArARRvU)
- 4. C++ locale 一例 (/articles/meIBjyv)
- 5. 考不上三本也能给自己心爱的语言加上Coroutine（四） (/articles/eiAzeiZ)
- 6. 女神经之 C++ 自学之路 （关于句柄） (/articles/AZZBvuI)

我来评几句

请输入评论内容...

登录后评论

已发表评论数(0)

相关站点

博客园
cnblogs.com

博客园-原创精华区 (/sites/Fn2umm)

＋订阅

热门文章

- 1. 突破GIL..... - 转载 - Quicklib 源码分析 3、4 (/articles/32eeUnv)
- 2. C语言中数组与指针的关系 (/articles/j6f6BbV)
- 3. 突破GIL？！ - 转载 - Quicklib 源码分析 1 (/articles/ArARRvU)
- 4. C++ locale 一例 (/articles/meIBjyv)
- 5. 考不上三本也能给自己心爱的语言加上Coroutine（四） (/articles/eiAzeiZ)