blowfish加密解密算法使用说明：

在项目中添加blowfish.h blowfish.cpp,在自己代码中实现加密接口和解密接口，实现如下：

//blowfish 解密

//传入参数：合作方的token,要解密的字符编码

//传出参数：解密后的字符串

void blowfish\_decode(const string& token, const string& en\_price, string& price)

{

CBlowFish oBlowFish((unsigned char\*)token.c\_str(), token.size());

string::size\_type lsize=((en\_price.size()+7)/8)\*8;

char\* src\_buf, \*dst\_buf;

if (0 == en\_price.size())

{

return;

}

src\_buf = new char[lsize];

dst\_buf = new char[lsize];

memset(src\_buf, 0x00, lsize);

memset(dst\_buf, 0x00, lsize);

memcpy(src\_buf, en\_price.c\_str(),en\_price.size());

oBlowFish.Decrypt((unsigned char\*)src\_buf, (unsigned char\*)dst\_buf, lsize, CBlowFish::ECB);

price = string(dst\_buf, lsize);

delete src\_buf;

delete dst\_buf;

}

//blowfish 加密

//传入参数：合作方的token,要加密的字符串

//传出参数：加密后的字符编码

void blowfish\_encode(const string& token, const string& price, string& en\_price)

{

CBlowFish oBlowFish((unsigned char\*)token.c\_str(), token.size());

string::size\_type lsize=((price.size()+7)/8)\*8;

char\* src\_buf, \*dst\_buf;

if (0 == price.size())

{

return;

}

src\_buf = new char[lsize];

dst\_buf = new char[lsize];

memset(src\_buf, 0x00, lsize);

memset(dst\_buf, 0x00, lsize);

memcpy(src\_buf, price.c\_str(),price.size());

oBlowFish.Encrypt((unsigned char\*)src\_buf, (unsigned char\*)dst\_buf, lsize, CBlowFish::ECB);

en\_price = string(dst\_buf, lsize);

delete src\_buf;

delete dst\_buf;

}

直接调用即可

示例如：test.cpp

执行：g++ test.cpp blowfish.cpp即可生成可执行文件a.out