Cryptopp使用方法

**1、测试MD5**

|  |  |
| --- | --- |
| 01 | #define CRYPTOPP\_ENABLE\_NAMESPACE\_WEAK 1 |
| 02 | #include "md5.h" |
| 03 | **using** **namespace** CryptoPP; |
| 04 | #pragma comment(lib, "cryptopp\\lib\\cryptlib.lib") |
| 05 |  |
| 06 | **using** **namespace** std; |
| 07 |  |
| 08 | **void** main() { |
| 09 |  |
| 10 | byte message[]="HelloWorld!"; |
| 11 | byte mres[16];//MD5 128 bits=16bytes |
| 12 |  |
| 13 | Weak::MD5 md5; |
| 14 | md5.Update(message,11);//strlen=11 |
| 15 | md5.Final(mres); |
| 16 |  |
| 17 | **for**(**int** i=0;i<16;i++) |
| 18 | **printf**("%02X",mres[i]); |
| 19 |  |
| 20 | **printf**("\n"); |
| 21 | } |

**2、测试AES**

|  |  |
| --- | --- |
| 01 | //For AES encrypt |
| 02 | #include "default.h" |
| 03 | #include "cryptlib.h" |
| 04 | #include "filters.h" |
| 05 | #include "bench.h" |
| 06 | #include "osrng.h" |
| 07 | #include "hex.h" |
| 08 | #include "modes.h" |
| 09 | #include "files.h" |
| 10 |  |
| 11 | **using** **namespace** CryptoPP; |
| 12 | #pragma comment(lib, "cryptopp\\lib\\cryptlib.lib") |
| 13 |  |
| 14 | **using** **namespace** std; |
| 15 |  |
| 16 | **void** main() { |
| 17 |  |
| 18 | unsigned **char** key[] = {0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08, 0x01,0x02, 0x03,0x04,0x05,0x06,0x07,0x08};//AES::DEFAULT\_KEYLENGTH |
| 19 | unsigned **char** iv[]  = {0x01,0x02,0x03,0x03,0x03,0x03,0x03,0x03, 0x03,0x03, 0x01,0x02,0x03,0x03,0x03,0x03}; |
| 20 | **int** keysize = 16; |
| 21 |  |
| 22 | string  message = "Hello World!"; |
| 23 | string  strEncTxt; |
| 24 | string  strDecTxt; |
| 25 |  |
| 26 | //CBC - PADDING |
| 27 | //AES-CBC Encrypt(ONE\_AND\_ZEROS\_PADDING) |
| 28 | CBC\_Mode<AES>::Encryption  Encryptor1(key,keysize,iv); |
| 29 | StringSource(   message, |
| 30 | **true**, |
| 31 | **new** StreamTransformationFilter( Encryptor1, |
| 32 | **new** StringSink( strEncTxt ), |
| 33 | BlockPaddingSchemeDef::BlockPaddingScheme::ONE\_AND\_ZEROS\_PADDING, |
| 34 | **true**) |
| 35 | ); |
| 36 |  |
| 37 | //AES-CBC Decrypt |
| 38 | CBC\_Mode<AES>::Decryption Decryptor1(key,keysize,iv); |
| 39 | StringSource(   strEncTxt, |
| 40 | **true**, |
| 41 | **new** StreamTransformationFilter( Decryptor1, |
| 42 | **new** StringSink( strDecTxt ), |
| 43 | BlockPaddingSchemeDef::BlockPaddingScheme::ONE\_AND\_ZEROS\_PADDING, |
| 44 | **true**) |
| 45 | ); |
| 46 |  |
| 47 |  |
| 48 | //CTR - NO\_PADDING |
| 49 | //AES-CTR Encrypt |
| 50 | CTR\_Mode<AES>::Encryption  Encryptor2(key,keysize,iv); |
| 51 | StringSource(   message, |
| 52 | **true**, |
| 53 | **new** StreamTransformationFilter( Encryptor2, |
| 54 | **new** StringSink( strEncTxt ), |
| 55 | BlockPaddingSchemeDef::BlockPaddingScheme::NO\_PADDING, |
| 56 | **true**) |
| 57 | ); |
| 58 |  |
| 59 | //AES-CTR Decrypt |
| 60 | CTR\_Mode<AES>::Decryption Decryptor2(key,keysize,iv); |
| 61 | StringSource(   strEncTxt, |
| 62 | **true**, |
| 63 | **new** StreamTransformationFilter( Decryptor2, |
| 64 | **new** StringSink( strDecTxt ), |
| 65 | BlockPaddingSchemeDef::BlockPaddingScheme::NO\_PADDING, |
| 66 | **true**) |
| 67 | ); |
| 68 |  |
| 69 | } |

上文是使用StringSource方式加密字符串，还可以使用FileSource方式直接对文件进行加解密操作。示例如下：

|  |  |
| --- | --- |
| 01 | SecByteBlock HexDecodeString(**const** **char** \*hex) { |
| 02 | StringSource ss(hex, **true**, **new** HexDecoder); |
| 03 | SecByteBlock result((**size\_t**)ss.MaxRetrievable()); |
| 04 | ss.Get(result, result.size()); |
| 05 | **return** result; |
| 06 | } |
| 07 |  |
| 08 | **void** AES\_CTR\_Encrypt(**const** **char** \*hexKey, **const** **char** \*hexIV, **const** **char** \*infile, **const** **char** \*outfile) { |
| 09 | SecByteBlock key = HexDecodeString(hexKey); |
| 10 | SecByteBlock iv = HexDecodeString(hexIV); |
| 11 |  |
| 12 | CTR\_Mode<AES>::Encryption aes(key, key.size(), iv); |
| 13 |  |
| 14 | FileSource(infile, **true**, **new** StreamTransformationFilter(aes, **new** FileSink(outfile))); |
| 15 | } |

直接调用AES\_CTR\_Encrypt函数即可，CBC函数需对应修改。

**3、StringSource类定义filters.h**

|  |  |
| --- | --- |
| 1 | //! zero terminated string as source |
| 2 | StringSource(**const** **char** \*string, **bool** pumpAll, BufferedTransformation \*attachment = NULL) |
| 3 | : SourceTemplate<StringStore>(attachment) {SourceInitialize(pumpAll, MakeParameters("InputBuffer", ConstByteArrayParameter(string)));} |
| 4 | //! binary byte array as source |
| 5 | StringSource(**const** byte \*string, **size\_t** length, **bool** pumpAll, BufferedTransformation \*attachment = NULL) |
| 6 | : SourceTemplate<StringStore>(attachment) {SourceInitialize(pumpAll, MakeParameters("InputBuffer", ConstByteArrayParameter(string, length)));} |
| 7 | //! std::string as source |
| 8 | StringSource(**const** std::string &string, **bool** pumpAll, BufferedTransformation \*attachment = NULL) |
| 9 | : SourceTemplate<StringStore>(attachment) {SourceInitialize(pumpAll, MakeParameters("InputBuffer", ConstByteArrayParameter(string)));} |

**4、RSA有关的加解密、签名函数**

更多请参考工程cryptest工程下test.cpp文件内函数

view source

print

?

|  |  |
| --- | --- |
| 1 | **void** GenerateRSAKey(unsigned **int** keyLength, **const** **char** \*privFilename, **const** **char** \*pubFilename, **const** **char** \*seed); |
| 2 | string RSAEncryptString(**const** **char** \*pubFilename, **const** **char** \*seed, **const** **char** \*message); |
| 3 | string RSADecryptString(**const** **char** \*privFilename, **const** **char** \*ciphertext); |
| 4 | **void** RSASignFile(**const** **char** \*privFilename, **const** **char** \*messageFilename, **const** **char** \*signatureFilename); |
| 5 | **bool** RSAVerifyFile(**const** **char** \*pubFilename, **const** **char** \*messageFilename, **const** **char** \*signatureFilename); |