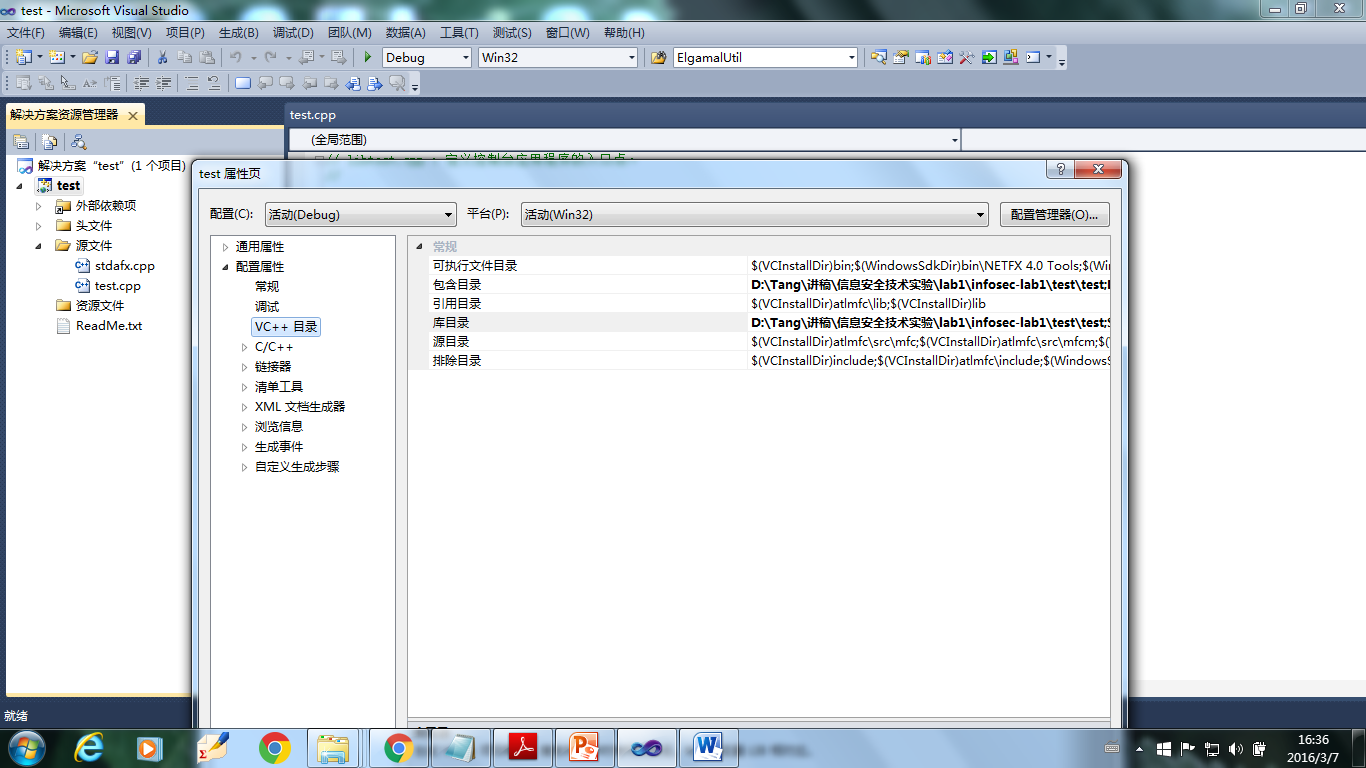
Crypto.lib库的安装与使用

安装环境：Windows 7 + Visual Studio 2010

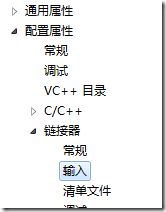
一 配置(以test为例)

1. 双击VC++控制台应用程序test

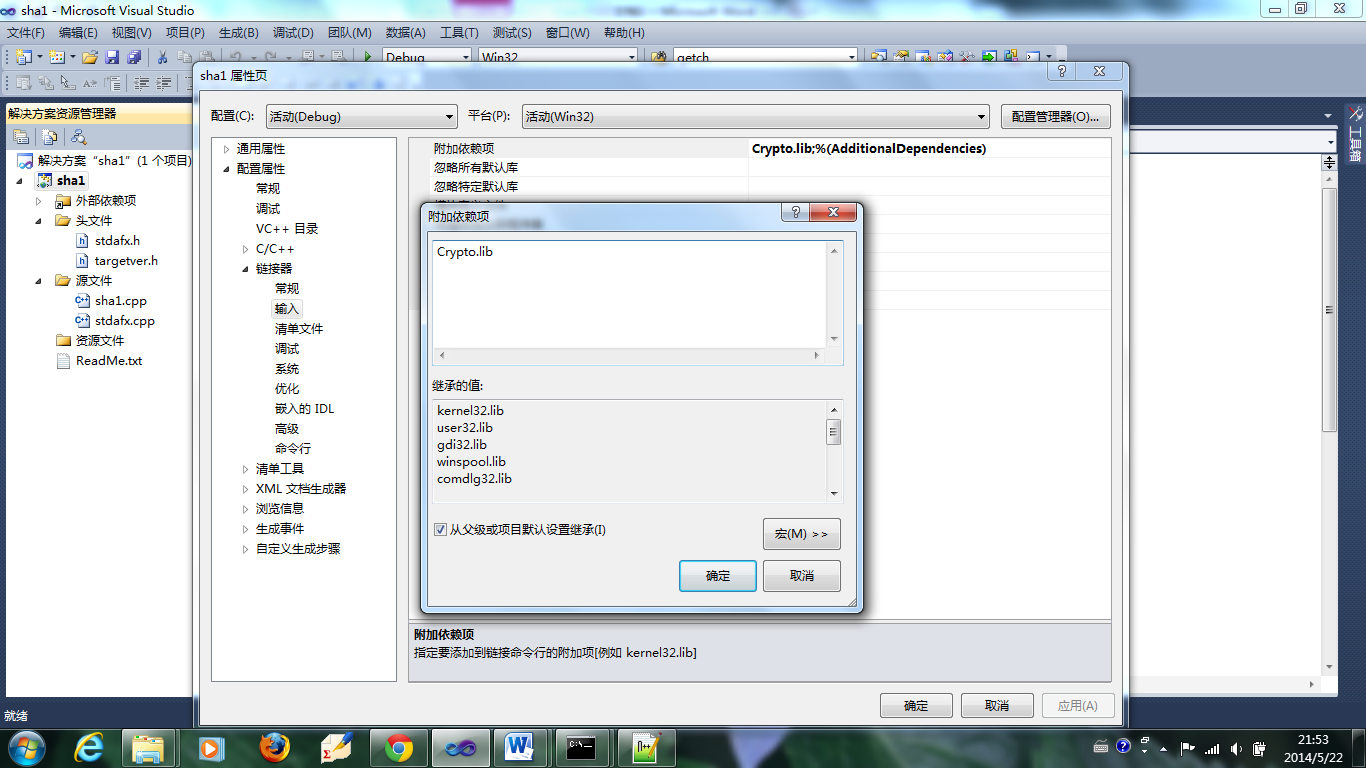
2．在项目名称中点击“右键”，选择“属性”，从里面找到“VC++目录”，设置库目录：库目录指向Crypto.lib所在的目录;包含目录指向openssl目录所在的目录。



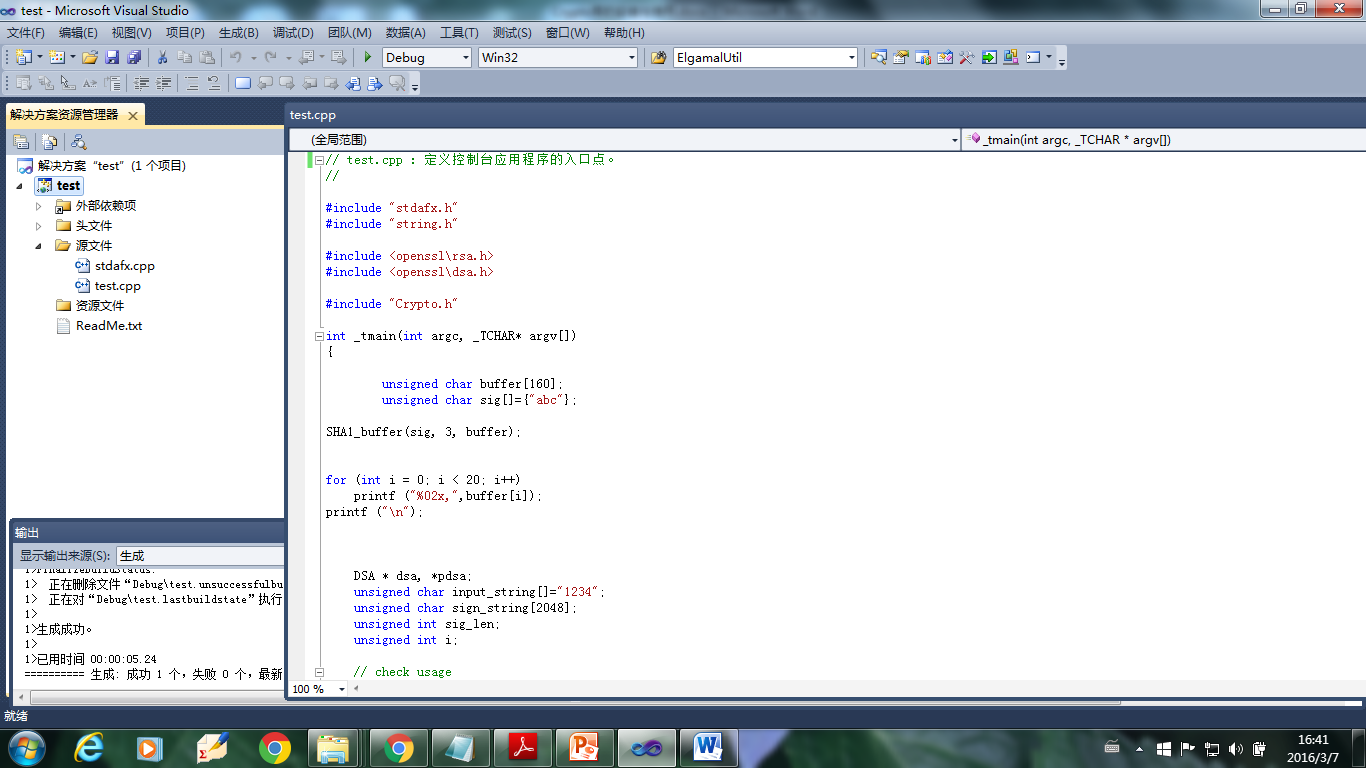
3. 切换到“链接器”-> “输入”：



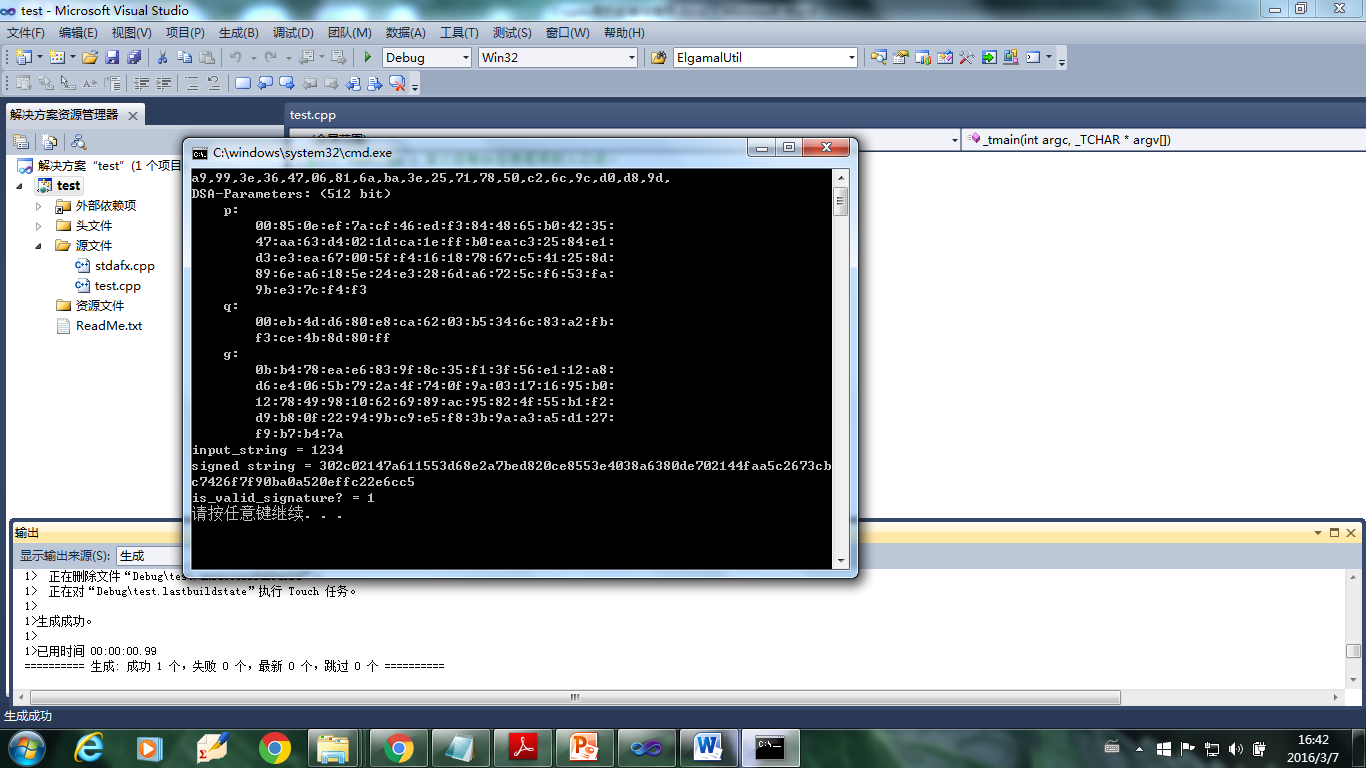
在“附加依赖项”中，加入Crypto.lib文件：



4. 引用



5.运行



二 函数接口

本接口函数在openssl基础上开发，需要使用openssl中的头文件。

1.Crypto.lib 有关SHA1的函数接口

#define SHA\_DIGEST\_LENGTH 20

void SHA1\_buffer(unsigned char \* buffer, int bufferlength, unsigned char \*outputSHA1Value);

void SHA1\_file(char \* filename, unsigned char \*outputSHA1Value);

2．Crypto.lib 有关AES的函数接口和宏

#define AES\_BLOCK\_SIZE 16

#define AES\_ENCRYPT 1

#define AES\_DECRYPT 0

void AES\_CBC128\_encrypt\_buffer (unsigned char \*in, unsigned char \*out, unsigned int length, unsigned char \*key, unsigned char \*iv);

void AES\_CBC128\_decrypt\_buffer (unsigned char \*in, unsigned char \*out, unsigned int length, unsigned char \*key, unsigned char \*iv);

void AES\_CTR128\_encrypt\_buffer (unsigned char \*in, unsigned char \*out, unsigned int length, unsigned char \*key, unsigned char \*counter);

void AES\_CTR128\_decrypt\_buffer (unsigned char \*in, unsigned char \*out, unsigned int length, unsigned char \*key, unsigned char \*counter);

void AES\_ECB128\_encrypt\_buffer(unsigned char \*in, unsigned char \*out, unsigned int length, unsigned char \*key);

void AES\_ECB128\_decrypt\_buffer(unsigned char \*in, unsigned char \*out, unsigned int length, unsigned char \*key);

void AES\_CFB128\_encrypt\_buffer (unsigned char \*in, unsigned char \*out, unsigned int length, unsigned char \*key, unsigned char \*iv);

void AES\_CFB128\_decrypt\_buffer (unsigned char \*in, unsigned char \*out, unsigned int length, unsigned char \*key, unsigned char \*iv);

void AES\_OFB128\_encrypt\_buffer (unsigned char \*in, unsigned char \*out, unsigned int length, unsigned char \*key, unsigned char \*iv);

void AES\_OFB128\_decrypt\_buffer (unsigned char \*in, unsigned char \*out, unsigned int length, unsigned char \*key, unsigned char \*iv);

说明：

1. 上述函数针对12位密钥的AES的五种工作模式，即ECB、CBC、CTR、CFB和OFB下的对给定缓冲区数据的AES加密；
2. 各个参数说明

in： 待加/解密数据缓冲区；

out： 已加/解密数据缓冲区；

length：待加/解密数据长度；

key： 128位AES密钥；

iv： 128位初始向量

1. Crypto.lib 有关RSA的函数接口

需要使用 <openssl\rsa.h>中的函数定义

void RSA\_init\_keypair (int bits, RSA \*\* publickey, RSA \*\* privatekey);

void RSA\_free\_keypair (RSA \* publickey, RSA \* privatekey);

void RSA\_public\_encrypt\_buffer (unsigned char \*in, int insize, unsigned char \*out, int \*outsize, RSA \*key);

void RSA\_private\_decrypt\_buffer (unsigned char \*in, int insize, unsigned char \*out, int \*outsize, RSA \*key);

void RSA\_private\_encrypt\_buffer (unsigned char \*in, int insize, unsigned char \*out, int \*outsize, RSA \*key);

void RSA\_public\_decrypt\_buffer (unsigned char \*in, int insize, unsigned char \*out, int \*outsize, RSA \*key);