
Subject: Re: Using openssl functions on U++
Posted by Zardos **on** Wed, 12 Dec 2007 21:00:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

ealabarce wrote on Wed, 12 December 2007 21:11Hello again, well im reading the documentation, and now i can get the md5 and the base64 conversion, but now I need to sign the md5 with a rsa .key private file, I trying to understand who to do this, viewing on the openssl.org API documentation, but I some confuse, let me explain:

First I initilize a RSA struture like this:

```
RSA *FirmaDigital;
```

Then i use the FileIn Stream:

```
FileIn StreamArchivoFirma;
```

Get a new Struct of RSA

```
FirmaDigital=RSA_new();
```

My cuestion is, who to put the file stream on the memory structure of RSA.

Thanks in advance.

I have attached a Upp package "CryptOpenSsl.zip".

There is a class with the following intefarce:

```
struct Rsa : public Moveable<Rsa> {  
    Rsa() { rsa = NULL; }  
    ~Rsa() { if(rsa) RSA_free(rsa); }
```

```
void GenerateKeyPair(int bits = 1024, int exponent = 17);
```

```
String PrivateKeyToPem();  
String PublicKeyToPem();  
void PrivateKeyFromPem(const String &pem);  
void PrivateKeyFromPem(uint8 *d, int l);  
void PublicKeyFromPem(const String &pem);  
void PublicKeyFromPem(uint8 *d, int l);
```

```
String SignSHA(const String &msg);  
String Decrypt(const String &msg, int padding = RSA_PKCS1_OAEP_PADDING);
```

```
bool VerifySHA(const String &msg, const String &sig);  
String Encrypt(const String &msg, int padding = RSA_PKCS1_OAEP_PADDING);
```

```

int MaxMsgCount(int padding = RSA_PKCS1_OAEP_PADDING);

void Serialize(Stream &s);

protected:
RSA *rsa;
};

a example:
#ifndef _DEBUG
TEST(Rsa) {
Rsa rsa;
rsa.GenerateKeyPair(512);

String pri = rsa.PrivateKeyToPem();
String pub = rsa.PublicKeyToPem();

Rsa rsa2;
rsa2.PrivateKeyFromPem(pri);

String pri2 = rsa2.PrivateKeyToPem();
String pub2 = rsa2.PublicKeyToPem();

CHECK(pri == pri2);
CHECK(pub == pub2);
CHECK(rsa.VerifySHA("Kleiner Test", rsa2.SignSHA("Kleiner Test")));
CHECK(rsa.Decrypt(rsa2.Encrypt("Kleiner Test")) == "Kleiner Test");
CHECK(!rsa.VerifySHA("Kleiner Test", rsa2.SignSHA("@Kleiner Test")));
}
#endif

```

Use "SignSHA" and "VerifySHA" for signing and verification.
 "Encrypt" and "Decrypt" for crypting.

The functions "PrivateKeyToPem" "PublicKeyToPem" "PrivateKeyFromPem"
 "PublicKeyFromPem" are useful, too. Use them to store or load a public / private key.

Just look in the cpp file how it is done and copy what you need or tweak the class for your requirements.

- Ralf

File Attachments

-
- 1) [CryptOpenSsl.zip](#), downloaded 442 times
 - 2) [UnitTest.zip](#), downloaded 404 times
-