老谷自言自语
我想、我行、我思，我将尽我所能

ios下使用rsa算法与php进行加解密通讯

Posted on 2011-10-27 by york_gu

首先了解一下几个相关概念，以方便后面遇到的问题的解决：

- RSA算法：1977年由Ron Rivest、Adi Shamirh和LenAdleman发明的，RSA就是取自他们三个人的名字。算法基于一个数论：将两个大素数相乘非常容易，但要对这个乘积的结果进行因式分解却非常困难，因此可以把乘积公开作为公钥。该算法能够抵抗目前已知的所有密码攻击。RSA算法是一种非对称算法，算法需要一对密钥，使用其中一个加密，需要使用另外一个才能解密。我们在进行RSA加密通讯时，就把公钥放在客户端，私钥留在服务器。
- DER, PEM：既然使用RSA需要一对密钥，那么我们当然是要先使用工具来生成这样一对密钥了。在linux、unix下，最简单方便的就是使用openssl命令行了。而DER、PEM就是生成的密钥可选择的两种文件格式。DER是Distinguished Encoding Rules的简称，是一种信息传输语法规则，在ITU X.690中定义的。在ios端，我们的公钥就是需要这样一种格式的，我们可以从Certificate, Key, and Trust Services Reference这篇文档的SecCertificateCreateWithData函数的data参数的说明中看到。而PEM格式是一种对DER进行封装的格式，他只是把der的内容进行了base64编码并加上了头尾说明。openssl命令行默认输出的都是PEM格式的文件，要能够在ios下使用，我们需要指定使用DER或者先生成PEM然后转换称DER。

使用openssl命令行生成密钥对

```
openssl req -x509 -out public_key.der -outform der -new -newkey rsa:1024 -keyout priva
```

按照提示，填入私钥的密码，签名证书的组织名、邮件等信息之后，就会生成包含有公钥的证书文件public_key.der合私钥文件private_key.pem。public_key.der文件用于分发到ios客户端进行公钥加解密，而private_key.pem文件留在服务器端供php使用。当然，如果为了在服务器端进行加解密测试，那么我们还可以生成一个服务器端PHP使用的pem公钥文件：

```
openssl rsa -in private_key.pem -pubout -out public_key.pem
```

上面这个命令就会根据输入的私钥文件生成pem格式的公钥文件了。从这里也可以看到，根据私钥，我们是可以生成相对应的公钥的，这也就是为什么我们要把公钥放在客户端，而不是私钥放在客户端的原因了。

服务器端PHP的加解密函数

闲话不多说，贴一段代码，肯定能看懂的了

```
class RSAEncryptTest {
  const PRIVATE_KEY = "-----BEGIN ENCRYPTED PRIVATE KEY-----
MIICxjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQI2aEhi35m/scCAggA
MBQGCCqGSIb3DQMHBAihDOZw68gfKwSCAoAu2E/d2a9FgHDWoinhK2nMc2MlrgL+
kpWcZ5YyUEWUw87DFKrG7dkpAYOgLIpyDatXUVFy2EekZH0Iplqo+yswtho8NtpJ
7T7KJ0nbUXo4we658Ez0EHAnWw+xZegsmJGk2+5QRCALDFyYEIMp3UvqxBjPjfDM
1rEZ0j2o9U40ouDqUVxTpq7ZwHkx/EkB8xHwKpFexz8J0s6gjPy6yLUjX2ut63LD
6X4YPBQLCJIcaLZORoAQ01cxCaM+78WTLUjdhcaFvff9f1xkiUU3XrQQTpuM/3YH
MQ6SMYDAgiOLqSCiMc0VABwf0/kdBnxu9/C/CK82ehA29cVAe8o7HgKg+WszCzTE
+QRCJ2fa7nOd7UXzCDfKh5Hhq1RjLFocVK8OW7tIgW3ircltM1ow30FfEzIdvzmv
LP0QhfGI3o9VT7r5qihGxtXtnGeUEGwvK0j0ozznfsNej7sVFP0Jfw39TdUlEENh
OPjtuBBBHv/oafQ3jqYnrI4R12ZrEU0acm85vRJm32K1RT1ROMFpc5sU20S8nMGC
I3iCzUlJPQF0t07bKexayvfWlJVAwEqBBCPTnvfTMBEt33iC72dQELbzMAM/n7th
TcY/sReO/J4beGk3//c7PImKIOcIvKF9Gp99l/+BM/LMZ7Thd/qwMOV6Eb3T4BvY
ItC+P5Lr29XeINmLRHXKwr27uTxX0fwDpmpwkPbGreVXA2cCxHnEzkh2WP3qGa7q
+Cwi03ISTEcZbNxLRGArtFUOIvNpz4+FS07OLWVKGl6K6bTffBx1tlZ492SqyNAC
7aP4/4I9Malnt0VjRKYPBCkTvVhoWBG+ThoOav5IV+w7ZDy8mtcrcAII
-----END ENCRYPTED PRIVATE KEY-----";
  const PUBLIC_KEY = "-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDAqjJs08oHvNdhlWC+kGBd90PD
7CVjClhRTk3nn+2NNaP4Bi5N/A18rdrV6clNAGUz4i/5q/VQXeLiGYYqgmAkKCJe
gReMsfcnoOSWu+Tvxih/48pu1hwBrmMLFZPOOUWQ9YjQEo7SYBe0HKoEl6XMqNwz
HV7sk9x6BKz9QeLi5QIDAQAB
-----END PUBLIC KEY-----";
  private static $private_key;
  private static $public_key;

  public static function private_encrypt($str){
    self::setup_key();
    if(openssl_private_encrypt($str, $encrypted, self::$private_key))
      return $encrypted;
  }
  public static function private_decrypt($str){
    self::setup_key();
    if(openssl_private_decrypt($str, $decrypted, self::$private_key))
      return $decrypted;
  }
  public static function public_decrypt($str){
    self::setup_key();
    if(openssl_public_decrypt($str, $decrypted, self::$public_key))
      return $decrypted;
  }
  public static function public_encrypt($str){
    self::setup_key();
    if(openssl_public_encrypt($str, $encrypted, self::$public_key))
      return $encrypted;
  }
  private static function setup_key(){
    if (!self::$private_key){
      // 这里的test就是在生成证书的时候设置的私钥密码
      self::$private_key = openssl_pkey_get_private(self::PRIVATE_KEY, "test");
    }
    if (!self::$public_key)
```

```
54        self::$public_key = openssl_pkey_get_public(self::PUBLIC_KEY);
55    }
56  }
```

IOS客户端的加解密

首先我们需要导入Security.framework，在ios中，我们主要关注四个函数

- **SecKeyEncrypt**：使用公钥对数据进行加密
- **SecKeyDecrypt**：使用私钥对数据进行解密
- **SecKeyRawVerify**：使用公钥对数字签名和数据进行验证，以确认该数据的来源合法性。什么是数字签名，可以参考百度百科这篇文章？
- **SecKeyRawSign**：使用私钥对数据进行摘要并生成数字签名

从这几个函数中，我们可以看到，我们使用公钥能做的事情就有两个：加密数据，以及对服务器端发来的数据进行签名认证，但是如果你想跟我之前想的一样，要使用公钥来对数据进行解密，那就没有自带API了。如果想在服务器端使用私钥加密数据，然后再在客户端使用公钥进行解密，以图这样来对交互数据进行加密，看来是行不通的。其实也应该是这样，公钥是公开的，因为他可以编译到二进制文本里面就认为他不能被获取其实是不对的。同时，RSA因为都是做大数的运算，算法性能上比较差，如果做大数据量的加解密，对IOS来讲，肯定也是不合适的。

这里就把使用公钥进行加密的代码贴出来：

```
1  // 我们在前面使用openssl生成的public_key.der文件的base64值，用你自己的替换掉这里
2  #define RSA_KEY_BASE64 @"MIIC5DCCAk2gAwIBAgIJALUk4hrYth9oMA0GCSqGSIb3DQEBBQUAMIGKMQsw
3  DTjERMA8GA1UECAwIU2hhbmdoYWkxETAPBgNVBAcMCFNoYW5naGFpMQ4wDAYDVQQKDAVCYWl5aTEOMAwGA1UE
4  xEDAOBgNVBAMMB1lvcmsuR3UxIzAhBgkqhkiG9w0BCQEWFGd5cTUzMTk5MjBAZ21haWwuY29tMB4XDTExMTAy
5  XDTExMTEyNTAyNDUzM1owgYoxCzAJBgNVBAYTAkNOMREwDwYDVQQIDAhTaGFuZ2hhaTERMA8GA1UEBwwIU2hh
6  MBgNVBAoMBUJhaXlpMQ4wDAYDVQQLDAVCYWl5aTEQMA4GA1UEAwwHWW9yay5HdTEjMCEGCSqGSIb3DQEJARYU
7  yMEBnbWFpbC5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAK3cKya7oOi8jVMkRGVuNn/SiSS1y5kn
8  DJZqo30LVPXXL9nHcYXBTulJgzutCOGQxw8ODfAKvXYxmX7QvLwlJRFEzrqzi3eAM2FYtZZeKbgV6PximOwCG
9  ezP1B2eWKz4kLIuSUKOmt0h3RpIPkatPBAgMBAAGjUDBOMB0GA1UdDgQWBBSIiLi2mehEgi/MwRZOld1mLlhl
10 EGDAWgBSIiLi2mehEgi/MwRZOld1mLlhl7TAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4GBAB0GUsss
11 DzNr8pB0idfI+Farl46OZnW5ZwPu3dvSmhQ+yRdh7Ba54JCyvRy0JcWB+fZgO4QorNRbVVbBSuPg6wLzPuasy
12 Iena6Z60aFWRwhazd6+hIsKTMTExaWjndblEbhAsjdpg6QMsKurs9+izr"
13
14 static SecKeyRef _public_key=nil;
15 + (SecKeyRef) getPublicKey{ // 从公钥证书文件中获取到公钥的SecKeyRef指针
16   if(_public_key == nil){
17     NSData *certificateData = [Base64 decode:RSA_KEY_BASE64];
18     SecCertificateRef myCertificate =  SecCertificateCreateWithData(kCFAllocatorDefau
19     SecPolicyRef myPolicy = SecPolicyCreateBasicX509();
20     SecTrustRef myTrust;
21     OSStatus status = SecTrustCreateWithCertificates(myCertificate,myPolicy,&myTrust)
22     SecTrustResultType trustResult;
23     if (status == noErr) {
24       status = SecTrustEvaluate(myTrust, &trustResult);
25     }
26     _public_key = SecTrustCopyPublicKey(myTrust);
27     CFRelease(myCertificate);
28     CFRelease(myPolicy);
29     CFRelease(myTrust);
30   }
31   return _public_key;
32 }
33
34 + (NSData*) rsaEncryptString:(NSString*) string{
35   SecKeyRef key = [self getPublicKey];
36   size_t cipherBufferSize = SecKeyGetBlockSize(key);
37   uint8_t *cipherBuffer = malloc(cipherBufferSize * sizeof(uint8_t));
38   NSData *stringBytes = [string dataUsingEncoding:NSUTF8StringEncoding];
39   size_t blockSize = cipherBufferSize - 11;
40   size_t blockCount = (size_t)ceil([stringBytes length] / (double)blockSize);
41   NSMutableData *encryptedData = [[[NSMutableData alloc] init] autorelease];
42   for (int i=0; i<blockCount; i++) {
43     int bufferSize = MIN(blockSize,[stringBytes length] - i * blockSize);
44     NSData *buffer = [stringBytes subdataWithRange:NSMakeRange(i * blockSize, bufferS
45     OSStatus status = SecKeyEncrypt(key, kSecPaddingPKCS1, (const uint8_t *)[buffer b
46                                     [buffer length], cipherBuffer, &cipherBufferSize)
47     if (status == noErr){
48       NSData *encryptedBytes = [[NSData alloc] initWithBytes:(const void *)cipherBuff
49       [encryptedData appendData:encryptedBytes];
50       [encryptedBytes release];
51     }else{
52       if (cipherBuffer) free(cipherBuffer);
53       return nil;
54     }
55   }
56   if (cipherBuffer) free(cipherBuffer);
57 //  NSLog(@"Encrypted text (%d bytes): %@", [encryptedData length], [encryptedData de
58 //  NSLog(@"Encrypted text base64: %@", [Base64 encode:encryptedData]);
59   return encryptedData;
60 }
```

This entry was posted in PHP, iOS and tagged ios, openssh, rsa. Bookmark the permalink.

21 Responses to *ios下使用rsa算法与php进行加解密通讯*

leavingme *says:*
2012-01-19 at 10:45 am

你好！
在使用第一个命令行生成的public_key.der，经过base64编码后，和下面C代码的public_key.der文件的长度值不一样。请问是什么原因呢？谢谢！

Reply

york_gu *says:*
2012-01-30 at 2:36 pm

在C代码中，为了代码美观，对BASE64字符串进行了换行处理，每行的结尾处都添加了反斜杠，不知道是不是这个原因

Reply

leavingme *says:*
2012-02-05 at 10:49 pm

hi，已经解决了～
解决的关键应该是第一句命令行缺少-x509参数。
openssl req -x509 -out public_key.der -outform der -new -newkey rsa:1024 -keyout private_key.pem

Reply

york_gu *says:*
2012-02-06 at 11:28 am

非常感谢网友leavingme的提醒

Reply

Anonymous *says:*
2013-04-24 at 2:55 pm

[Base64 decode:RSA_KEY_BASE64]
这方法从哪来的?

Reply

york_gu *says:*
2013-05-07 at 9:41 am

iOS上的Base64编码库，你可以搜索一下

Reply

york_gu *says:*
2013-07-14 at 11:46 am

可以在网上搜一下Base64的ios编码库，有些库的函数调用方法可能不是这样，不过也没有关系，只要能够执行base64解码就OK了

Reply

---

chang *says:*
2012-03-07 at 3:53 pm

你好，我想请问下define RSA_KEY_BASE64 这个值和我从服务器那边直接拿到的publickey是同一个值么，我把我的publickey放进去，运行SecCertificateRef myCertificate 出错，生不成变量，我想是不是publickey需要经过什么编码处理呢，我这边现在没有der等证书，只有同学给的publickey，谢谢了

Reply

york_gu *says:*
2012-03-07 at 7:43 pm

public key是否能在iOS代码中正确运行取决于你的public key是不是符合der的编码规范，你同学给你的public key可能是另外一种格式编码方式的base64字符串，这样的话就肯定解析不出来。要解决你的这个问题，你需要跟你的同学沟通他是怎么获取到这个public key的。

Reply

Robbie *says:*
2012-10-26 at 9:16 am

我正在做相关的项目，看到你的POST实在是太感动了！

Reply

Gil *says:*
2012-11-06 at 2:38 pm

Java 代码如何获取符合der的编码规范的public key给iOS客户端使用?

Reply

Gil *says:*
2012-11-06 at 2:39 pm

KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("RSA");
keyPairGen.initialize(1024);
KeyPair keyPair = keyPairGen.generateKeyPair();
PublicKey publicKey = (RSAPublicKey) keyPair.getPublic();
PrivateKey privateKey = (RSAPrivateKey) keyPair.getPrivate();

这样生成的，Base64传到iOS上时，无法还原成SecCertificateRef

Reply

york_gu *says:*
2012-11-06 at 5:22 pm

这个还真不清楚，不过openssl生成的公钥用java也是可以使用的

Reply

Anonymous *says:*
2012-12-24 at 3:23 pm

你好，请问下要如何将encryptedData转换成string？我的服务器端也是php的。
Encrypted text (size: 128 bytes):
这是我NSLog出来的结果，因为我要post去服务器端，所以我要的encryptedData大概是
dXXEEADvRDdk4QVM6TU9dyYtuHe8RVzP3KsWAoDDSQ2w+NELVPOv4mSVW1jIhEhb+pLsmOvLzGIW1/2qkp91M1UoPuhgplV2JEgP8EvcBn8EY43n7wVASGP9heRKolYMfENP54f35+GF
这样的。

我是参考了http://blog.iamzsx.me/show.html?id=155002

不好意思，我是个新手。

Reply

> **york_gu** *says:*
> 2012-12-26 at 7:46 pm
>
> 可以用base64编码
>
> Reply

---

leetvin *says:*
2013-04-14 at 10:14 pm

你好 问一下，这个开一个修改为支持java的bc那种生成不变的rsa值么。。

Reply

Terrence *says:*
2013-06-14 at 1:45 pm

我这里是pem格式的Base64 会报错 请问如何转成你der格式的编码?

Reply

---

ssp *says:*
2013-07-08 at 10:18 am

您好 请问 我的公钥是服务器给nsstring类型的 怎么用您的方法呢?

Reply

Tomby *says:*
2014-03-05 at 8:25 pm

I have encrypted the data that is from iOS. But the decoded text include some strange string besides right text behind.

Reply

---

silentcloud *says:*
2014-04-18 at 10:26 am

想请教一下，如果用 private key string 来进行 iOS 端解密?

Reply

求教 *says:*
2014-05-27 at 10:31 am

大哥好，请问能把那个der的base64值怎么来得？可以的话能把加密和解密的代码给我一份吗

Reply
