

## openssl 非对称加密算法RSA命令详解

## 1、非对称加密算法概述

非对称加密算法也称公开密钥算法，其解决了对称加密算法密钥分配的问题，非对称加密算法基本特点如下：

- 1、加密密钥和解密密钥不同
- 2、密钥对中的一个密钥可以公开
- 3、根据公开密钥很难推算出私人密钥

根据非对称加密算法的特点，可用户数字签名、密钥交换、数据加密。但是由于非对称加密算法较对称加密算法加密速度慢很多，故最常用的用途是数字签名和密钥交换。

目前常用的非对称加密算法有RSA、DH和DSA三种，但并非都可以用于密钥交换和数字签名。而是RSA可用于数字签名和密钥交换，DH算法可用于密钥交换，而DSA算法专门用于数字签名。

openssl支持以上三种算法，并为三种算法提供了丰富的指令集，本章主要介绍RSA算法及相关指令

## 2、RSA算法相关指令及用法

RSA虽然可以数字签名、密钥交换和数据加密，但是RSA加密数据速度慢，通常不使用RSA加密数据。所以最常用的功能就是数字签名和密钥交换，抛开数字签名和密钥交换的概念，实质上就是使用公钥加密还是使用私钥加密的区别。所以我们只要记住一句话：“公钥加密，私钥签名”。

公钥加密：用途是密钥交换，用户A使用用户B的公钥将少量数据加密发送给B，B用自己的私钥解密数据

私钥签名：用途是数字签名，用户A使用自己的私钥将数据的摘要信息加密一并发送给B，B用A的公钥解密摘要信息并验证

openssl中RSA算法指令主要有三个，其他指令虽有涉及，但此处不再详述。

指令	功能
genrsa	生成并输入一个RSA私钥
rsa	处理RSA密钥的格式转换等问题
rsautl	使用RSA密钥进行加密、解密、签名和验证等运算

### 2.1 genrsa指令说明

genrsa用于生成密钥对，其用法如下

```
xlzh@cmos:~$ openssl genrsa -
usage: genrsa [args] [numbits] //密钥
位数, 建议1024及以上
-des encrypt the generated key with DES in cbc mode //生成
的密钥使用des方式进行加密
-des3 encrypt the generated key with DES in ede cbc mode (168 bit key) //生成
的密钥使用des3方式进行加密
-seed encrypt PEM output with cbc seed //生成
的密钥还是要seed方式进行
-aes128, -aes192, -aes256 encrypt PEM output with cbc aes //生成
的密钥使用aes方式进行加密
-camellia128, -camellia192, -camellia256 encrypt PEM output with cbc camellia //生成
的密钥使用camellia方式进行加密
-out file output the key to 'file' //生成
的密钥文件, 可从中提取公钥
```

访问人数：

UV

访问总量：

PV

昵称： Gordon0918

园龄： 6年8个月

粉丝： 21

关注： 1

+加关注

2020年4月						
<	一	二	三	四	五	六
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

## 搜索

## 常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

## 我的标签

android(6)  
逆向(4)  
Hook(2)  
ida(2)  
genrsa(2)  
RSA(2)  
smali(2)  
so(1)  
sphinx(1)  
substrate(1)  
更多

## 随笔分类

android(6)  
android安全(11)  
C/C++(2)  
git(1)  
Linux(3)  
openssl(8)  
Scrapy(2)  
Windows(1)  
渗透测试  
协议(2)

随笔档案

- 2017年12月(1)
- 2017年4月(6)
- 2017年3月(4)
- 2016年6月(4)
- 2016年5月(1)
- 2016年4月(5)
- 2016年3月(6)
- 2016年1月(5)
- 2015年7月(1)
- 2015年1月(3)
- 2014年7月(1)

最新评论

- 1. Re:openssl 对称加密算法enc命令详解  
fedora 29 x86 workstation OpenSSL 1.1.1d FIPS 10 Sep 2019 没有 aes-256-gcm. openssl enc -ciphers 何解... --NickD
- 2. Re:openssl 对称加密算法enc命令详解  
-pass env:passwd 的passwd的前面不需要加\$ ? --crazyloser
- 3. Re:Android AccessibilityService(辅助服务) 使用示例  
他是返回的整个activity 的view , 所以会包含三个Fragment 的 --伍歌歌
- 4. Re:PPTP协议握手流程分析  
大佬 自己能软件模拟vpn 并建立通道 进行数据传输吗 --54辉哥
- 5. Re:Https协议简析及中间人攻击原理  
写的不错 --aqu415

阅读排行榜

- 1. openssl 对称加密算法enc命令详解(25875)
- 2. openssl 证书请求和自签名命令req详解(23152)
- 3. 如何把java代码转换成smali代码(20907)
- 4. openssl 非对称加密算法RSA命令详解(18579)
- 5. openssl 摘要和签名验证指令dgst使用详解(18023)

评论排行榜

- 1. 如何把java代码转换成smali代码(4)
- 2. android调试系列--使用ida pro调试原生程序(3)
- 3. openssl 对称加密算法enc命令详解(2)
- 4. openssl 非对称加密算法RSA命令详解(1)
- 5. openssl 非对称加密算法DSA命令详解(1)

推荐排行榜

- 1. openssl 证书请求和自签名命令req详解(5)
- 2. Android调试系列—使用android studio调试smali代码(3)
- 3. openssl 非对称加密算法RSA命令详解(2)
- 4. openssl CA服务器模拟指令CA详解(1)
- 5. Https协议简析及中间人攻击原理(1)

```
-passout arg      output file pass phrase source           //指定
密钥文件的加密码令, 可从文件、环境变量、终端等输入

-f4               use F4 (0x10001) for the E value         //选择
指数e的值, 默认指定该项, e值为65537 -3                  use 3 for the E value
//选择指数e的值, 默认值为65537, 使用该选项则指数指定为3

-engine e         use engine e, possibly a hardware device. //指定
三方加密库或者硬件

-rand file:file:...
                  load the file (or the files in the directory) into //产生
随机数的种子文件

                  the random number generator
```

可以看到genrsa指令使用较为简单, 常用的也就有指定加密算法、输出密钥文件、加密码令。我们仅举一个例子来说明

```
/*
 * 指定密钥文件rsa.pem
 * 指定加密算法aes128
 * 指定加密密钥123456
 * 指定密钥长度1024
 */
xlzh@cmos:~$ openssl genrsa -out rsa.pem -aes128 -passout pass:123456 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001) // 默认模式65537
/*加密后的密钥文件有加密算法等信息*/
xlzh@cmos:~$ cat rsa.pem
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,4C23682B0D34D339ED7E44819A70B4F9

c9uHqQWbkcw3hjdQ/6fGuJcOfchd4+KfVZoJnnISnJBAhv3CelFAksKb2Rka5GoC
4Eq6SykCCSH8OboPoPBjd1ZdAsDl1Pio0vIJfAoQ4NmaRJ6l+6onJ/HAX2NFTDjN
yrmsGOWejB6A3MT4KiXrvICnkKMsUYlQp6ln2qOeVynmxeWAWiVZnjfm0OkScLlK
RGsuL32vecN5b1S8ftZYJTS3PQxjmyaw65zLx+8mUObanL9WhSLtZ2eo/6xTzRbD
iOGMolFP/3ObqIAS3007qV48CtwWr1Aa+RpbMViESN7BforOaNbh0s5NVuUnXYs
hx90izj2M1L4i5SP8jKBunXPk6CHQtUQXpMH06nhoMNYZPtRQegFgZlwVOpOfos5
khGAjJpNEXI7ah8oCNYO2lJV6SlMFxK1lUeS3xCvM8Cd/zVBSzD7jg+axBJr+LpO
rhpmEFkStXhtFo30K3BoyQHIzYEH4S59xWO+dfrb2zUvkKsQKkV+TFMSZpr7b7U
iegUcK3NrbcWDApfTYmf/edublJBv816to+hYQLhXKfuzP5iMjmnubhrXrA6S47
7XN6ni19DGWzUEMPnH6Brc8mj7JwFtxdpWDN2pY+VcJ04098f008c+4eSS3u0Y9f
TyxYy1C9nIwxF+t2Dulq94N4AQ2uyTXoVNhrmDYrJ9BUCug6gz6xtU24aSGFvtn
ikGAU8JcX0GkcwU60tTLsXPNawhNxJSJ5n7BXaV6QQlGOiikQlJAcRv2PMxNqVgK
poVq742+awsichrwQE5VIFW9AdSMYIT7w06IogyUrS+0+FmFS6qPtT3ZbFZakzkd
-----END RSA PRIVATE KEY-----
xlzh@cmos:~$
```

2.2 rsa指令说明

rsa指令用户管理生成的密钥, 其用法如下

```
xlzh@cmos:~$ openssl rsa -
unknown option -
rsa [options] <infile >outfile
where options are
-inform arg      input format - one of DER NET PEM           //输入文件格式, 默认
pem格式
-outform arg     output format - one of DER NET PEM           //输入文件格式, 默认
pem格式
-in arg         input file                                     //输入文件
-sgckey         Use IIS SGC key format                        //指定SGC编码格式,
兼容老版本, 不应再使用
-passin arg     input file pass phrase source                 //指定输入文件的加密
口令, 可来自文件、终端、环境变量等
-out arg       output file                                     //输出文件
-passout arg    output file pass phrase source                 //指定输出文件的加密
口令, 可来自文件、终端、环境变量等
-des           encrypt PEM output with cbc des                 //使用des加密输出的
文件
-des3          encrypt PEM output with ede cbc des using 168 bit key //使用des3加密输出的
文件
```

```

-seed            encrypt PEM output with cbc seed           //使用seed加密输出的
文件
-aes128, -aes192, -aes256
                    encrypt PEM output with cbc aes         //使用aes加密输出的
文件
-camellia128, -camellia192, -camellia256
                    encrypt PEM output with cbc camellia     //使用camellia加密
输出的文件呢
-text            print the key in text                       //以明文形式输出各个
参数值
-noout           don't print key out                         //不输出密钥到任何文
件
-modulus         print the RSA key modulus                   //输出模数指
-check          verify key consistency                       //检查输入密钥的正确
性和一致性
-pubin          expect a public key in input file           //指定输入文件是公钥
-pubout         output a public key                         //指定输出文件是公钥
-engine e       use engine e, possibly a hardware device.   //指定三方加密库或者
硬件
xlzh@cmos:~$

```



rsa指令操作示例如下

#### 1、rsa添加和去除密钥的保护口令

```

/*生成不加密的RSA密钥*/
xlzh@cmos:~/test$ openssl genrsa -out RSA.pem
Generating RSA private key, 512 bit long modulus
.....+++++++
.....+++++++
e is 65537 (0x10001)
/*为RSA密钥增加口令保护*/
xlzh@cmos:~/test$ openssl rsa -in RSA.pem -des3 -passout pass:123456 -out E_RSA.pem
writing RSA key
/*为RSA密钥去除口令保护*/
xlzh@cmos:~/test$ openssl rsa -in E_RSA.pem -passin pass:123456 -out P_RSA.pem
writing RSA key
/*比较原始后的RSA密钥和去除口令后的RSA密钥，是一样*/
xlzh@cmos:~/test$ diff RSA.pem P_RSA.pem

```



#### 2、修改密钥的保护口令和算法

```

/*生成RSA密钥*/
xlzh@cmos:~/test$ openssl genrsa -des3 -passout pass:123456 -out RSA.pem
Generating RSA private key, 512 bit long modulus
.....+++++++
.....+++++++
e is 65537 (0x10001)
/*修改加密算法为aes128，口令是123456*/
xlzh@cmos:~/test$ openssl rsa -in RSA.pem -passin pass:123456 -aes128 -passout
pass:123456 -out E_RSA.pem
writing RSA key

```



#### 3、查看密钥对中的各个参数

```
xlzh@cmos:~/test$ openssl rsa -in RSA.pem -des -passin pass:123456 -text -noout
```

#### 4、提取密钥中的公钥并打印模数值

```

/*提取公钥，用pubout参数指定输出为公钥*/
xlzh@cmos:~/test$ openssl rsa -in RSA.pem -passin pass:123456 -pubout -out pub.pem
writing RSA key
/*打印公钥中模数值*/
xlzh@cmos:~/test$ openssl rsa -in pub.pem -pubin -modulus -noout
Modulus=C35E0B54041D78466EAE7DE67C1DA4D26575BC1608CE6A199012E11D10ED36E2F7C651D4D8B40D936
91D901E2CF4E21687E912B77DCCE069373A7F6585E946EF

```

#### 5、转换密钥的格式

```

/*把pem格式转化成der格式，使用outform指定der格式*/
xlzh@cmos:~/test$ openssl rsa -in RSA.pem -passin pass:123456 -des -passout pass:123456 -

```

```
outform der -out rsa.der
writing RSA key
/*把der格式转化成pem格式，使用inform指定der格式*/
xlzh@cmos:~/test$ openssl rsa -in rsa.der -inform der -passin pass:123456 -out rsa.pem
```


2.3 rsautl指令说明

上述两个指令是密钥的生成及管理作用，rsautl则是真正用于密钥交换和数字签名。实质上就是使用RSA公钥或者私钥加密。


而无论是使用公钥加密还是私钥加密，RSA每次能够加密的数据长度不能超过RSA密钥长度，并且根据具体的补齐方式不同输入的加密数据最大长度也不一样，而输出长度则总是跟RSA密钥长度相等。RSA不同的补齐方法对应的输入输入长度如下表

数据补齐方式	输入数据长度	输出数据长度	参数字符串
PKCS#1 v1.5	少于(密钥长度-11)字节	同密钥长度	-pkcs
PKCS#1 OAEP	少于(密钥长度-11)字节	同密钥长度	-oaep
PKCS#1 for SSLv23	少于(密钥长度-11)字节	同密钥长度	-ssl
不使用补齐	同密钥长度	同密钥长度	-raw

rsautl指令用法如下



```
xlzh@cmos:~$ openssl rsautl -
Usage: rsautl [options]
-in file          input file                //输入文件
-out file         output file               //输出文件
-inkey file       input key                 //输入的密钥
-keyform arg     private key format - default PEM //指定密钥格式
-pubin           input is an RSA public    //指定输入的是RSA公钥
-certin          input is a certificate carrying an RSA public key //指定输入的是证书文件
-ssl             use SSL v2 padding         //使用SSLv23的填充方式
-raw             use no padding            //不进行填充
-pkcs            use PKCS#1 v1.5 padding (default) //使用V1.5的填充方式
-oaep            use PKCS#1 OAEP           //使用OAEP的填充方式
-sign            sign with private key     //使用私钥做签名
-verify          verify with public key    //使用公钥认证签名
-encrypt         encrypt with public key   //使用公钥加密
-decrypt         decrypt with private key  //使用私钥解密
-hexdump         hex dump output          //以16进制dump输出
-engine e        use engine e, possibly a hardware device. //指定三方库或者硬件设备
-passin arg      pass phrase source       //指定输入的密码
```



rsautl操作示例如下：

1、使用rsautl进行加密和解密操作



```
/*生成RSA密钥*/
xlzh@cmos:~/test$ openssl genrsa -des3 -passout pass:123456 -out RSA.pem
Generating RSA private key, 512 bit long modulus
.....++++++
..++++++
e is 65537 (0x10001)
/*提取公钥*/
xlzh@cmos:~/test$ openssl rsa -in RSA.pem -passin pass:123456 -pubout -out pub.pem
writing RSA key
/*使用RSA作为密钥进行加密，实际上使用其中的公钥进行加密*/
xlzh@cmos:~/test$ openssl rsautl -encrypt -in plain.txt -inkey RSA.pem -passin
pass:123456 -out enc.txt
/*使用RSA作为密钥进行解密，实际上使用其中的私钥进行解密*/
xlzh@cmos:~/test$ openssl rsautl -decrypt -in enc.txt -inkey RSA.pem -passin pass:123456
-out replain.txt
/*比较原始文件和解密后文件*/
xlzh@cmos:~/test$ diff plain.txt replain.txt
/*使用公钥进行加密*/
xlzh@cmos:~/test$ openssl rsautl -encrypt -in plain.txt -inkey pub.pem -pubin -out
enc1.txt
```

```
/*使用RSA作为密钥进行解密，实际上使用其中的私钥进行解密*/
xlzh@cmos:~/test$ openssl rsautl -decrypt -in enc1.txt -inkey RSA.pem -passin pass:123456
-out replain1.txt
/*比较原始文件和解密后文件*/
xlzh@cmos:~/test$ diff plain.txt replain1.txt
```



在进行这个实验的时候有个疑惑，为什么相同的明文，使用密钥加密和公钥加密后的密文结果不一样？在网上查询了下，是因为rsa公钥加密的时候根据填充模式填充随机数，导致每次加密结果不同。

## 2、使用rsautl进行签名和验证操作

```
/*提取PKCS8格式的私钥*/
xlzh@cmos:~/test$ openssl pkcs8 -topk8 -in RSA.pem -passin pass:123456 -out pri.pem -
nocrypt
/*使用RSA密钥进行签名，实际上使用私钥进行加密*/
xlzh@cmos:~/test$ openssl rsautl -sign -in plain.txt -inkey RSA.pem -passin pass:123456 -
out sign.txt
/*使用RSA密钥进行验证，实际上使用公钥进行解密*/
xlzh@cmos:~/test$ openssl rsautl -verify -in sign.txt -inkey RSA.pem -passin pass:123456
-out replain.txt
/*对比原始文件和签名解密后的文件*/
xlzh@cmos:~/test$ diff plain.txt replain.txt
/*使用私钥进行签名*/
xlzh@cmos:~/test$ openssl rsautl -sign -in plain.txt -inkey pri.pem -out sign1.txt
/*使用公钥进行验证*/
xlzh@cmos:~/test$ openssl rsautl -verify -in sign1.txt -inkey pub.pem -pubin -out
replain1.txt
/*对比原始文件和签名解密后的文件*/
xlzh@cmos:~/test$ cat plain replain1.txt
```



要注意这里的签名和验证过程其本质上是加解密操作，不是标准意义上的签名和验证。标准意义上签名和验证是需要增加摘要操作的，后续文章再详细阐述。

## 3、小结

我们可以看到上述指令的参数中有涉及到证书相关的内容，等到后期我们介绍CA相关内容的时候在进行补充。

分类: [openssl](#)

标签: [RSA](#), [非对称加密](#), [genrsa](#), [rsautl](#)

好文要顶

关注我

收藏该文



[Gordon0918](#)

[关注 - 1](#)

[粉丝 - 21](#)

[+加关注](#)

2

0

« 上一篇: [openssl AES加密算法API的使用示例](#)

» 下一篇: [openssl 非对称加密算法DSA命令详解](#)

posted @ 2016-04-07 17:16 [Gordon0918](#) 阅读(18579) 评论(1) 编辑 收藏

### 评论

#1楼 2017-06-09 14:18 | JC31

赞!

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

【推荐】开发者必看：MVP时间线上峰会，技术进阶行业实战，让你快速成长！

【推荐】腾讯云产品限时秒杀，爆款1核2G云服务器99元/年！

#### 相关博文:

- 非对称加密过程详解（基于RSA非对称加密算法实现）
- openssl 非对称加密算法DSA命令详解
- 【Java】聊聊常用的非对称加密算法之一RSA的使用（Java）
- RSA加解密算法
- JAVA 非对称加密算法RSA

» 更多推荐...

#### 最新 IT 新闻:

- 新MacBook Air评测：性价比提升，同价位最值得买的苹果笔记本
- 新冠疫情影响美科技产业 初创企业3月累计裁员近4000人
- 受新冠病毒疫情影响 施乐宣布放弃对惠普“蛇吞象”式敌意收购
- 文件显示特斯拉拖着不愿关厂 卫生官员称是“公共健康风险”
- 传iPhone 9或4月22日开卖 代码曝光具备车钥匙功能

» 更多新闻...