链滴

从零开始 OpenSSL 之 (壹) - 使用 genrsa、rsa、rsautl 生成公私钥

作者：adlered

原文链接：https://ld246.com/article/1552806383042

来源网站：链滴

公钥和私钥

在生活中，我们常常会经历加密->解密的过程。当你在扫码支付、登录用户、进行游戏的时候，**大部分**据包传输的都是**加密后**的数据包。

俗话讲加密

尽我所能，我会将公钥和私钥的概念最简单地讲述出来。

假设A是客户端，B是服务端：

1. **A**使用某些手段生成了一段完整的私钥
2. **A**使用完整的私钥经过某种算法生成了另一段公钥
3. **A**将公钥和完整的私钥给了**B**

当**B**想要将东西传给**A**时：

1. **B**把公钥和要传送的文件乱七八糟地用某种算法**掺杂在一起**发送给**A**
2. 当　**B**发送给**A**时，如果有**黑客**拦截了这个文件，他只能获得**一堆乱码**
3. 当　**A**收到**一堆乱码**后，使用完整的私钥经过某些算法提取出**原文件**

由于**A**和**B**同时拥有公钥和私钥，所以双方都可以**使用公钥加密**并且**使用私钥解密**。

OpenSSL

原文链接：从零开始 OpenSSL 之 (壹) - 使用 genrsa、rsa、rsautl 生成公私钥

我们可以将OpenSSL理解为一个**工具箱**，它能实现大部分的密钥加、解密以及更多高级功能。

请注意

在下文中的命令有&&连接符，为的是让多条命令按顺序执行。

生成私钥

现在，让我们使用OpenSSL生成一段私钥。

在你的Linux或MacOS (Windows需自行安装)中执行下面命令：

```
mkdir keys && cd keys
```

为了方便整理思路，我们只在keys文件夹下进行操作。

```
openssl genrsa -out private.key 1024 && ls && cat private.key
```

我们使用openssl随机生成一串私钥并导出到private.key，然后列出当前文件夹下的内容可以看到private.key，最后抓取private.key中的内容：

```
Generating RSA private key, 1024 bit long modulus
.....................................................................++++++
.....................++++++
e is 65537 (0x10001)
private.key
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC1FTjWrrsokP6DwFL055b0LRdPq1UqcokS7V+jN2EBy79msaG7
CDOe7d8sgX8f8DYAXNQ58LGulfvFoCJp1Luoah7eYHcgBa5JiHD3wakkMsnveT8f
tfKziTLg7Yzes9mprA3afVvvH4UfLkAIkC437bCbaAe3E0YF1FmI0RdfrwIDAQAB
AoGAC6E2Sktkh8FwJyQF7+ajwkDXG23EhXpfpVCLcN8Qot3kCANmgK2RKybZy3Dx
qipyPWm405PKxYyqY9HaA/P+rUGeYmJHgx4tA/Q30uel6a2T1wFvYbo2b+FzBsLp
aXi/KVaxLNTx2h3g8gf8DjTt5deBJeW0ZtMp1O9NyFo2hLECQQDfvL6WoCBMYZBG
5M9PJ0IE9CriT91beCfUFn+s53umPsmYBjDde6hZ6f9jkXJ4uNt0crT2YP35Prfl
lLo00+wJAkEAzzHkYsC2wZ/MdiwNOyueNgO2xCmjczkoBuwTCgRtr/Yaal2xpU74
BhLT1Eld5JrXbmqE5POhd+XYqNcyPLhL9wJAEpJYP6iLBcwTbc8QZkGb+U1LQf3f
IfiXVcOIVZHhcatEkJny9G+PSedii6FHHz44+TGMU+c5LbdHDIeL7VaqAQJBAJj+
hV9qYEj/f5HvG2IK5kuKWIXnHTuzA9yAq9pkxL3/IyhRwSCI/Fo3wtVpy3xH3jY5
T6JkSGHphh++r+joSacCQAorZDk0lmEiZDkXcC44OmoO283FIGR5qUYMcoCX/AH5
5gCy3NOASL0YBfWw9oi93poIKzwKx/xgHa3VSLioYp4=
-----END RSA PRIVATE KEY-----
```

我们可以看到OpenSSL已经成功生成了一串私钥。

生成公钥

公钥是用来和原文件进行混淆传输的，它可以被理解为是私钥的不完整版。

输入以下命令：

```
openssl rsa -in private.key -pubout -out public.key && ls && cat public.key
```

-in 使用密钥

-pubout 输出公钥
-out 输出名称

openssl rsa的参数如下：

```
adler@localhost ~/keys: openssl rsa --help
Invalid cipher '-help'
usage: rsa [-ciphername] [-check] [-in file] [-inform fmt]
    [-modulus] [-noout] [-out file] [-outform fmt] [-passin src]
    [-passout src] [-pubin] [-pubout] [-sgckey] [-text]

 -check            Check consistency of RSA private key
 -in file          Input file (default stdin)
 -inform format     Input format (DER, NET or PEM (default))
 -modulus           Print the RSA key modulus
 -noout             Do not print encoded version of the key
 -out file          Output file (default stdout)
 -outform format    Output format (DER, NET or PEM (default PEM))
 -passin src        Input file passphrase source
 -passout src       Output file passphrase source
 -pubin             Expect a public key (default private key)
 -pubout            Output a public key (default private key)
 -sgckey            Use modified NET algorithm for IIS and SGC keys
 -text              Print in plain text in addition to encoded
```

本文我们使用了openssl genrsa、rsa、rsautl三种，分别用于生成密钥、加密文件和解密文件，你可输入openssl genrsa/rsa/rsautl --help获取帮助。

你会得到如下结果：

```
writing RSA key
private.key public.key
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1FTjWrrsokP6DwFL055b0LRdP
q1UqcokS7V+jN2EBy79msaG7CDOe7d8sgX8f8DYAXNQ58LGulfvFoCJp1Luoah7e
YHcgBa5JiHD3wakkMsnveT8ftfKziTLg7Yzes9mprA3afVvvH4UfLkAIkC437bCb
aAe3E0YF1FmI0RdfrwIDAQAB
-----END PUBLIC KEY-----
```

你会看到公钥实际上比私钥要少得多。

假装有第二台电脑

实际上我们将文件加密后，是传输到另一台电脑然后使用私钥进行解密的，但本次实验我们**略过**加密件的传输，直接进行加密和解密。

将文件加密

让我们新建一个文件，并加入内容：

```
echo 'Hello world!' > text.txt && cat text.txt
```

如果返回Hello world!证明你的文件创建完毕。

现在，让我们使用公钥对这个文件进行加密：

```
openssl rsautl -encrypt -in text.txt -inkey public.key -pubin -out encryptedFile.txt && cat encr
ptedFile.txt
```

你可以使用：

```
openssl rsautl --help
```

查看全部参数。

-encrypt 加密文件
-in 输入文件
-inkey 输入密钥
-pubin 表示使用-inkey输入的是公钥
-out 输出到指定文件

输入命令后，系统会返回已经加密的文本。

现在，假设这段文本已经传输到了另一台服务器上，并且该服务器拥有之前生成的密钥。

解密

当我们使用公钥加密文件并传输后，就需要使用之前获得到的私钥来解密。

如何解密，我们会在第二章详说。你可以**点击这里**跳转到第二章。