

openssl 对称加密算法enc命令详解

1、对称加密算法概述

openssl的加密算法库提供了丰富的对称加密算法，我们可以通过openssl提供的对称加密算法指令的方式使用，也可以通过调用openssl提供的API的方式使用。

openssl的对称加密算法指令主要用来对数据进行加密和解密处理，openssl基本上为所有其支持的对称加密算法都提供了指令的方式的应用，这些应用指令的名字基本上都是以对称加密算法本身的名字加上位数、加密模式或者其他属性组合而成。例如DES算法的CBC模式，其对应的指令就是des-cbc。可以通过命令查看当前版本的openssl支持的对称加密算法，例如Ubuntu14.04 openssl版本及支持对称加密算法指令如下：

```
xlzh@cmos:~$ openssl enc -help
unknown option '-'
options are
...
/**/
Cipher Types
-aes-128-cbc                -aes-128-cbc-hmac-sha1    -aes-128-cfb
-aes-128-cfb1              -aes-128-cfb8            -aes-128-ctr
-aes-128-ecb              -aes-128-gcm             -aes-128-ofb
-aes-128-xts              -aes-192-cbc             -aes-192-cfb
-aes-192-cfb1            -aes-192-cfb8            -aes-192-ctr
-aes-192-ecb             -aes-192-gcm             -aes-192-ofb
-aes-256-cbc             -aes-256-cbc-hmac-sha1   -aes-256-cfb
-aes-256-cfb1            -aes-256-cfb8            -aes-256-ctr
-aes-256-ecb             -aes-256-gcm             -aes-256-ofb
-aes-256-xts             -aes128                  -aes192
-aes256                  -bf                      -bf-cbc
-bf-cfb                  -bf-ecb                  -bf-ofb
-blowfish                -camellia-128-cbc        -camellia-128-cfb
-camellia-128-cfb1       -camellia-128-cfb8       -camellia-128-ecb
-camellia-128-ofb       -camellia-192-cbc        -camellia-192-cfb
-camellia-192-cfb1      -camellia-192-cfb8       -camellia-192-ecb
-camellia-192-ofb       -camellia-256-cbc        -camellia-256-cfb
-camellia-256-cfb1      -camellia-256-cfb8       -camellia-256-ecb
-camellia-256-ofb       -camellia128             -camellia192
-camellia256            -cast                    -cast-cbc
-cast5-cbc              -cast5-cfb              -cast5-ecb
-cast5-ofb              -des                     -des-cbc
-des-cfb                -des-cfb1               -des-cfb8
-des-ecb                -des-edc                -des-edc-cbc
-des-edc-cfb           -des-edc-ofb            -des-edc3
-des-edc3-cbc          -des-edc3-cfb           -des-edc3-cfb1
-des-edc3-cfb8         -desx                    -desx-cbc
-des3                   -id-aes128-GCM           -id-aes256-GCM
-rc2                    -rc2-40-cbc             -rc2-64-cbc
-rc2-cbc                -rc2-cfb                -rc2-ecb
-rc2-ofb                -rc4                     -rc4-40
-rc4-hmac-md5           -seed                   -seed-cbc
-seed-cfb              -seed-ecb               -seed-ofb
```

可以看到上述我们执行的是enc -help命令，enc是什么东西？原来openssl提供了两种方式调用对称加密算法：

一种就是直接调用对称加密指令，例如：

```
openssl des-cbc -in plain.txt -out encrypt.txt -pass pass:12345678
```

另外一种是使用enc的方式，即用对称加密指令作为enc指令的参数，例如：.

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 >						
日	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

<input type="text"/>	<input type="button" value="找找看"/>
<input type="text"/>	<input type="button" value="谷歌搜索"/>

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\(2\)](#)

```
openssl enc -des-cbc -in plain.txt -out encrypt.txt -pass pass:12345678
```

上述两条指令完成的功能是一样的，而且其参数也是一样。原来enc是作用是什么呢？简单来说，为了省事.....。

openssl提供了N多的对称加密算法指令，enc就是把这些N多的对称的加密算法指令统一集成到enc指令中。当用户使用时，只需使用enc，指定加密算法，就是完成单独的加密算法指令完成的操作。而且，enc中可以指定的对称加密算法指令可能并没有以单独指令的形式存在。所有笔者建议使用enc这种方式。

当然，虽然openssl为我们提供的对称加密算法指令虽然功能强大，但并不完整，例如对称加密算法不支持76位的RC2加解密或者84位的RC4加解密灯功能。如果想灵活的使用这些加密算法和模式，就需要学习openssl提供的API

2、对称加密算法指令参数

可以通过enc的man手册查看enc的详细用法，也可以通过enc -help的方式查看主要参数概要说明，如下



```
xlzh@cmos:~$ openssl enc -help
unknown option '-help'
options are
-in <file>      input file
-out <file>      output file
-pass <arg>      pass phrase source
-e              encrypt
-d              decrypt
-a/-base64      base64 encode/decode, depending on encryption flag
-k              passphrase is the next argument
-kfile          passphrase is the first line of the file argument
-md             the next argument is the md to use to create a key
                from a passphrase. One of md2, md5, sha or sha1
-S             salt in hex is the next argument
-K/-iv          key/iv in hex is the next argument
-[pP]          print the iv/key (then exit if -P)
-bufsize <n>    buffer size
-nopad          disable standard block padding
-engine e       use engine e, possibly a hardware device.
Cipher Types
...
```



[in/out]

这两个参数指定输入文件和输出文件，加密是输入文件是明文，输出文件是密文；解密时输入文件是密文，输出文件是明文。

[pass]

指定密码的输入方式，共有五种方式：命令行输入(stdin)、文件输入(file)、环境变量输入(var)、文件描述符输入(fd)、标准输入(stdin)。默认是标准输入，及从键盘输入。

[e/d]

e:加密， d:解密 默认是加密

[-a/-base64]

由于文件加密后是二进制形式，不方便查看，使用该参数可以使加密后的内容经过base64编码，使其可读；同样，解密时需要先进行base64解编码，然后进行解密操作。

[-k/-kfile]

兼容以前版本，指定密码输入方式，现已被pass参数取代

[md]

指定密钥生成的摘要算法，用户输入的口令不能直接作为文件加密的密钥，而是经过摘要算法做转换，此参数指定摘要算法，默认md5

[-S]

为了增强安全性，在把用户密码转换成加密密钥的时候需要使用盐值，默认盐值随机生成。使用该参数，则盐值由用户指定。也可指用 -nosalt 指定不使用盐值，但降低了安全性，不推荐使用。

[K/IV]

默认文件的加密密钥的Key和IV值是有用户输入的密码经过转化生成的，但也可以由用户自己指定Key/IV值，此时pass参数不起作用

[pP]

加上p参数会打印文件密钥Key和IV值，加上P参数也会打印文件密钥Key和IV值，但不进行真正的加解密操作

[bufsize]

读写文件的I/O缓存，一般不需要指定

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(25871)
2. openssl 证书请求和自签名命令req详解(23150)
3. 如何把java代码转换成smali代码(20906)
4. openssl 非对称加密算法RSA命令详解(18578)
5. openssl 摘要和签名验证指令dgst使用详解(18021)

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调试系列一使用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)

[-nopad]

不使用补齐，这就需要输入的数据长度是使用加密算法的分组大小的倍数

[engine]

指定三方加密设备，没有环境，暂不实验

3、对称加密算法使用示例

1、只对文件进行base64编码，而不使用加解密

```
/*对文件进行base64编码*/
openssl enc -base64 -in plain.txt -out base64.txt
/*对base64格式文件进行解密操作*/
openssl enc -base64 -d -in base64.txt -out plain2.txt
/*使用diff命令查看可知解码前后明文一样*/
diff plain.txt plain2.txt
```

2、不同方式的密码输入方式

```
/*命令行输入，密码123456*/
openssl enc -aes-128-cbc -in plain.txt -out out.txt -pass pass:123456
/*文件输入，密码123456*/
echo 123456 > passwd.txt
openssl enc -aes-128-cbc -in plain.txt -out out.txt -pass file:passwd.txt
/*环境变量输入，密码123456*/
passwd=123456
export passwd
openssl enc -aes-128-cbc -in plain.txt -out out.txt -pass env:passwd
/*从文件描述输入*/
openssl enc -aes-128-cbc -in plain.txt -out out.txt -pass fd:1
/*从标准输入输入*/
openssl enc -aes-128-cbc -in plain.txt -out out.txt -pass stdin
```

3、固定salt值加密

```
xlzh@cmos:~$ openssl enc -aes-128-cbc -in plain.txt -out encrypt.txt -pass pass:123456 -P
salt=32F5C360F21FC12D
key=D7E1499A578490DF940D99CAE2E29EB1
iv =78EEB538897CAF045F807A97F3CFF498
xlzh@cmos:~$ openssl enc -aes-128-cbc -in plain.txt -out encrypt.txt -pass pass:123456 -P
salt=DAA482697BECAB46
key=9FF8A41E4AC011FA84032F14B5B88BAE
iv =202E38A43573F752CCD294EB8A0583E7
xlzh@cmos:~$ openssl enc -aes-128-cbc -in plain.txt -out encrypt.txt -pass pass:123456 -P
-S 123
salt=1230000000000000
key=50E1723DC328D98F133E321FC2908B78
iv =1528E9AD498FF118AB7ECB3025AD0DC6
xlzh@cmos:~$ openssl enc -aes-128-cbc -in plain.txt -out encrypt.txt -pass pass:123456 -P
-S 123
salt=1230000000000000
key=50E1723DC328D98F133E321FC2908B78
iv =1528E9AD498FF118AB7ECB3025AD0DC6
xlzh@cmos:~$
```

可以看到，不使用-S参数，salt参数随机生成，key和iv值也不断变化，当salt值固定时，key和iv值也是固定的。

4、加解密后过程使用base64编解码

```
/*使用-a参数加密后使用base64编码*/
xlzh@cmos:~$ openssl enc -aes-128-cbc -in plain.txt -a -out encrypt.txt -pass pass:123456
/*使用-a参数解密前使用base64解码*/
xlzh@cmos:~$ openssl enc -aes-128-cbc -in encrypt.txt -d -a -out plain1.txt -pass
pass:123456
/*文件一样*/
xlzh@cmos:~$ diff plain.txt plain1.txt
/*加密后文件使用了base64编码*/
xlzh@cmos:~$ cat encrypt.txt
U2FsdGVkX19KbCj9GMI1TBOQjP8JJcefIUH1tHwf/Z4=
```

5、手动指定Key和IV值

```
/*手动指定key和iv值, salt固定*/
xlzh@cmos:~$ openssl enc -aes-128-cbc -in plain.txt -out encrypt.txt -K 1223 -iv f123 -p
salt=0B00000000000000
key=12230000000000000000000000000000
iv =F1230000000000000000000000000000
/*指定pass密码, 不起作用, 注意Key和IV值是16进制*/
xlzh@cmos:~$ openssl enc -aes-128-cbc -in plain.txt -out encrypt.txt -K 1223 -iv f123 -p
-pass pass:123456
salt=F502F4B8DE62E0E5
key=12230000000000000000000000000000
iv =F1230000000000000000000000000000
```

分类: [openssl](#)

标签: [openssl命令](#) [对称加密算法](#) [enc命令](#) [aes](#)

好文要顶

关注我

收藏该文



[Gordon0918](#)

[关注 - 1](#)

[粉丝 - 21](#)

[+加关注](#)

1

0

« 上一篇: [Linux UGO和ACL权限管理](#)

» 下一篇: [Linux能力\(capability\)机制的继承](#)

posted @ 2016-03-26 21:21 [Gordon0918](#) 阅读(25871) 评论(2) [编辑](#) [收藏](#)

评论

#1楼 2020-01-07 10:31 | [creazyloser](#)

-pass env:passwd 的passwd的前面不需要加\$?

[支持\(0\)](#) [反对\(0\)](#)

#2楼 2020-03-07 18:39 | [NickD](#)

fedora 29 x86 workstation OpenSSL 1.1.1d FIPS 10 Sep 2019 没有 aes-256-gcm.

`openssl enc -ciphers` 何解

[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】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:

- [openssl enc\(对称加密\)](#)
- [openssl 非对称加密算法RSA命令详解](#)
- [openssl 非对称加密算法RSA命令详解](#)
- [Openssl enc命令](#)

· [openssl enc 加解密](#)

» [更多推荐...](#)

最新 IT 新闻:

· [量子计算公司D-Wave宣布向任何COVID-19病毒研究者提供量子计算云服务](#)

· [《动物森友会》爆红背后，任天堂是这么「利用」人性的](#)

· [招募波音前高管担任负责人 亚马逊拟扩大无人机送货业务](#)

· [Mozilla正着手将Android上的Firefox Beta迁移到Fenix引擎](#)

· [SpaceX公布Starship手册 介绍它如何取代航天飞机并提供舒适旅途](#)

» [更多新闻...](#)