wang@wang-Lenovo:~$ gpg --gen-key

gpg (GnuPG) 1.4.16; Copyright (C) 2013 Free Software Foundation, Inc.

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:

(1) RSA and RSA (default)

(2) DSA and Elgamal

(3) DSA (sign only)

(4) RSA (sign only)

Your selection? 1

RSA keys may be between 1024 and 4096 bits long.

What keysize do you want? (2048) 2048

Requested keysize is 2048 bits

Please specify how long the key should be valid.

0 = key does not expire

<n> = key expires in n days

<n>w = key expires in n weeks

<n>m = key expires in n months

<n>y = key expires in n years

Key is valid for? (0) 0

Key does not expire at all

Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID

from the Real Name, Comment and Email Address in this form:

"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: hana-crosreg-key

Email address: cherish.wang@bitland.com.cn

Comment: wang bing

You selected this USER-ID:

"hana-crosreg-key (wang bing) <cherish.wang@bitland.com.cn>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o

You need a Passphrase to protect your secret key.

We need to generate a lot of random bytes. It is a good idea to perform

some other action (type on the keyboard, move the mouse, utilize the

disks) during the prime generation; this gives the random number

generator a better chance to gain enough entropy.

Not enough random bytes available. Please do some other work to give

the OS a chance to collect more entropy! (Need 288 more bytes)

gpg: key 2DDE9675 marked as ultimately trusted

public and secret key created and signed.

gpg: checking the trustdb

gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model

gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u

pub 2048R/2DDE9675 2016-10-12

Key fingerprint = 7D80 937C E2F3 EBB4 D654 A9D7 6674 E31A 2DDE 9675

uid hana-crosreg-key (wang bing) <cherish.wang@bitland.com.cn>

sub 2048R/7C1506FA 2016-10-12

wang@wang-Lenovo:~$ gpg --export -a hana-crosreg-key hana\_crosreg.pub

-----BEGIN PGP PUBLIC KEY BLOCK-----

Version: GnuPG v1

mQENBFf90hgBCADuWsDGJVQg3iNu0qksFhemU/3Fyei2t18GMomEUS/H9nu+sdzu

6BYCmV4L6c3TGlFnuNKwFNm3J+wyrp2bsYvR0l5FNY+u3prUfE9i40xMQ20c2/Gq

arPeoiikDOZc9HUuZ713gjpOk/BqT79V8szuq1Z3wKo7jMBqJBkshhxu7znH8O+0

D7PxjaN83tWbtEdnjtUvh566IT69xX38dxGXtgTTDfxvoUpL4pXQrnyizmvRJI/r

lI3c8x48W0aTKLK7SOH56FmQdUm3McZCkQc2djYTfHzGF31yq/qjni7vYRNh5ywb

8Fvzd2IgPQazfjMrHtwjeI5HoMIpnr6HBtmnABEBAAG0OmhhbmEtY3Jvc3JlZy1r

ZXkgKHdhbmcgYmluZykgPGNoZXJpc2gud2FuZ0BiaXRsYW5kLmNvbS5jbj6JATgE

EwECACIFAlf90hgCGwMGCwkIBwMCBhUIAgkKCwQWAgMBAh4BAheAAAoJEGZ04xot

3pZ1vQcH/RIn4aMySaYD9fARqUCT0PvCu92xqIK2lEkGafd9/7Kmsl1rWNRaAoSx

gIamWKwMnAnZrDASzEV/hnvRFn851X+dDrTYVugUI9MY5hj3Qme/EtRD1Je6Y5yb

nPRccsv1E5kpN41OBLlgARQusA69+8w3PdieYErcbqCx+cUn/WtsBeHsVUidqjCo

uDn+qzLfq+d9qCcjF7gTtmJ6HmoOe4BCB+VJs4CFV2LW6dHNP/sraQzxlNYTUGS5

Q5KwiiCNIty0id7cd4CQM44Hkp3XajG5Y71mA9HHEzdIgxgA+nUxw81LRinFTcjw

t0eVevjeV1CpWx0z2RWxgkrNhrQly4G5AQ0EV/3SGAEIAMgbbHntGC0DlJYMy603

Jb/wItp7qhssLcuuBJ3gOWWbtHlN2bwbop65YCMOpTxg7Eh68BdCHgOnf1s9iWhD

/ymOLkp3bwOWbydpKcr5sl8A8ArbcwWrTKsG+b2W2xExEsrpR8Md+LelPXo7dqh3

opd3jgXT/xpXWjlCX7rmGXWvxYXbzLAfauJK4WguJYswBEsdmSajgaQSBr0wo4Fq

O0f8dpU787ZVu/vdgSeekhi8gPdnFH/6dJzSESuLzvyZvfz3v+fRNlGjhBj4ET13

EPuSk26S+39/9SZQdxBTkbqf+u4Qm7IcjSoZv9wNeb5LMWhh+TaftrW7xojoA4KO

pbsAEQEAAYkBHwQYAQIACQUCV/3SGAIbDAAKCRBmdOMaLd6WdY7bB/4kIrHY5MWW

SqamdJI7I1ynr+JrDfC2m9hK7N3txaO5eWUxyWfZP7djQfnkrw4iB3HdT4LK90EZ

VELetpYEK7Py6Ne4whvxCnADk82hPejeGQa25vRQTYrc4aJhPfZY5IXDLBr1LMhi

Wg3jh1sD65ZBP9d2novBcha+4s0bE4WteUMF8K5YCtPca+0A/aKe3AzAGCoHVuVO

PBEC1G01igBgnWMnb8yMGiKyF/F5lwvcHOQI5sHlcQ4eeZf49SodRFN6T3TTn7JA

mf5nVSWzqYyW2K5Q3J4u6NNnEaG8eH09TuMch9r84Lq0TxcMrbEDcD1UsBi+Pr9K

tv6H6dtioeZr

=zoIG

-----END PGP PUBLIC KEY BLOCK-----

解码本地a89875656

gpg 讲解：<http://www.ruanyifeng.com/blog/2013/07/gpg.html>

$ gpg --edit-key 标识名

The problem is caused by the lack of entropy (or random system noise). While trying to get more, you might keep running into this message. In our case running a find on the disk, while making sha1sums and putting that into files, was actually not enough.

To check the available entropy, check the kernel parameters:

cat /proc/sys/kernel/random/entropy\_avail

36

To solve it, run the rngd command: rngd -f -r /dev/urandom

Debian / Ubuntu: apt-get install rng-tools

1、生成密钥对：gpg --gen-key  
   
为用户生成新密钥对。需提供：密钥类型（默认为RSA/RSA）；密钥长度（以位为单位，越长越强）；过期时间（以防密钥损坏）；（通常我都是一路回车过 去）接下来的还是要填一填的：名称、电子邮箱、标识密钥所有者的注释；密码短语（必须提供，如果私钥被盗，将无法使用）。  
   
2、列出公钥：gpg --list-keys  
   
列出所拥有的公钥：他们自己的公钥以及从与之通信的其他人那里导入的任何公钥。  
   
3、导出公钥：gpg --export --armor key-id -o file.key

gpg -o pbody\_crosreg.pub --export --armor pbody-crosreg-key   
   
将公钥导出至文件，以便于其他人使用。--armor选项以文本形式显示输出，而非二进制格式。key-id是电子邮箱地址或在--list-keys的pub行中列出的八位十六进制数。  
   
4、导入公钥：gpg --import file.key  
   
从发送给您的密钥文件中导入其他人的公钥  
   
5、[加密](http://www.2cto.com/Article/jiami/)文件：gpg --encrypt --armor -r key-id file

gpg --recipient google-crosreg-key --encrypt hana-5000codes-1000gsize-20161013T223759962

   
用key-id的公钥加密消息。如果未提供-r key-id，命令将提示收件人输入。默认输出文件为file.asc.  
   
6、解密文件：gpg --decrypt file

gpg -o yogi\_region.txt --decrypt Desktop/总结/Yogi/pbody-5000codes-1000gsize-20161031T005530241.gpg   
   
用您的私钥之一解密用公钥加密的消息。  
 