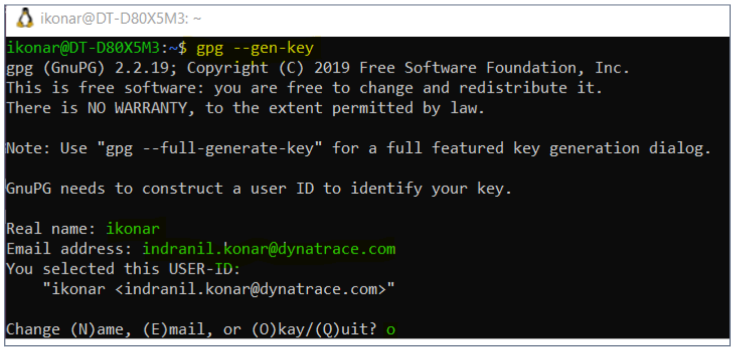
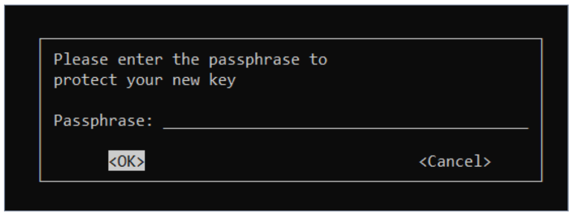
Encrypt and Decrypt using PGCRYPTO

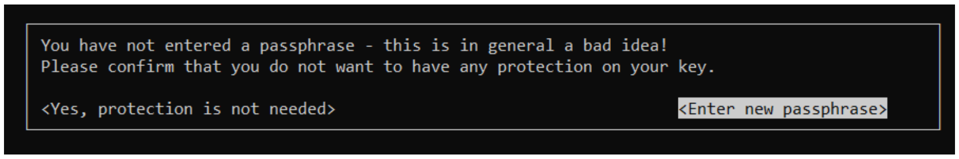
Step 1: Generate a private and public key.

- Install WSL if you have a window laptop.
- Open the terminal and type **gpg --gen-key**, provide user name and email address and type **o** (for OKAY) and hit enter.
- 

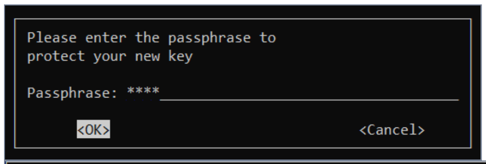
```
ikonar@DT-D80X5M3: ~
ikonar@DT-D80X5M3:~$ gpg --gen-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: ikonar
Email address: indranil.konar@dynatrace.com
You selected this USER-ID:
    "ikonar <indranil.konar@dynatrace.com>"

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

- On next screen, choose **OK** and hit enter.
- 

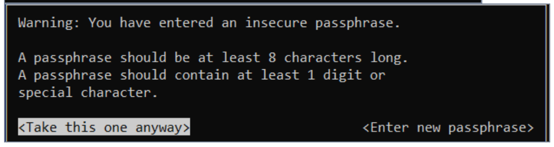
```
Please enter the passphrase to
protect your new key

Passphrase: ______________________________

      <OK>                        <Cancel>
```

- Choose to **Passphase** a below.
- 

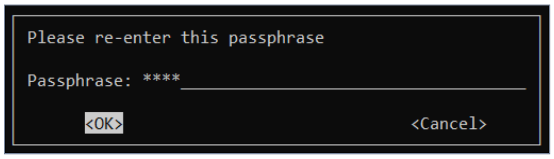
```
You have not entered a passphrase - this is in general a bad idea!
Please confirm that you do not want to have any protection on your key.

<Yes, protection is not needed>                   <Enter new passphrase>
```

- Enter your **Passphase**. I have used **abcd** for the demo. However as per below snap please choose a strong **Passphase**.
- 

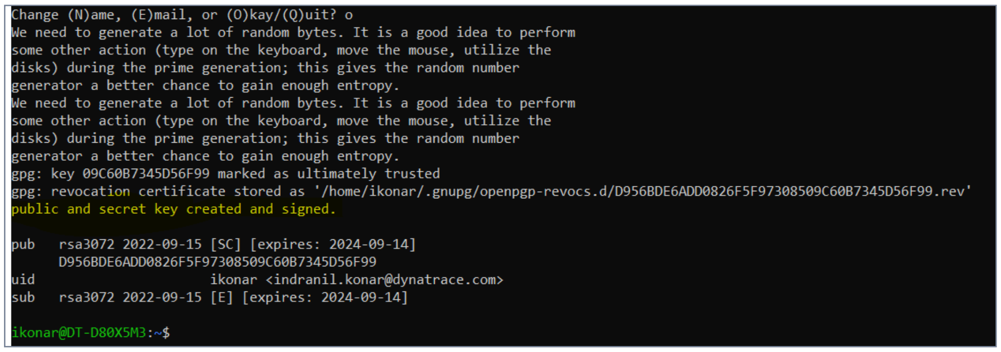
```
Please enter the passphrase to
protect your new key

Passphrase: ****__________________________

      <OK>                        <Cancel>
```



```
Warning: You have entered an insecure passphrase.

A passphrase should be at least 8 characters long.
A passphrase should contain at least 1 digit or
special character.

<Take this one anyway>                <Enter new passphrase>
```

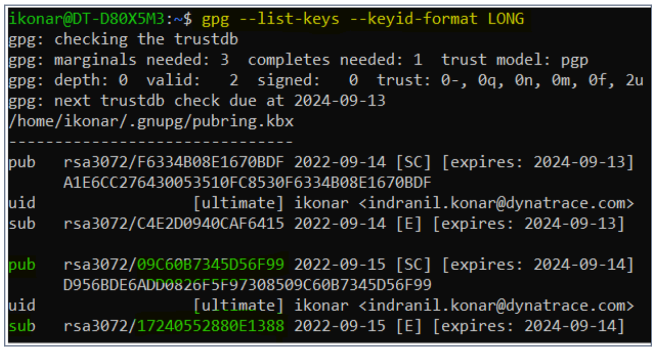
- Re-enter same **Passphrase** and chose **OK** and hit enter.
- 

```
Please re-enter this passphrase

Passphrase: ****__________________________

      <OK>                        <Cancel>
```

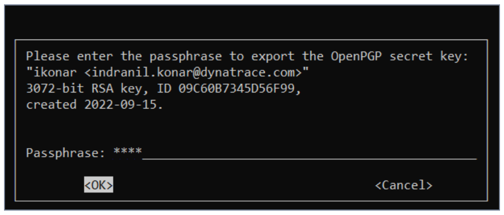
- 

```
Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
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.
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.
gpg: key 09C60B7345D56F99 marked as ultimately trusted
gpg: revocation certificate stored as '/home/ikonar/.gnupg/openpgp-revocs.d/D956BDE6ADD0826F5F97308509C60B7345D56F99.rev'
public and secret key created and signed.

pub   rsa3072 2022-09-15 [SC] [expires: 2024-09-14]
      D956BDE6ADD0826F5F97308509C60B7345D56F99
uid                      ikonar <indranil.konar@dynatrace.com>
sub   rsa3072 2022-09-15 [E] [expires: 2024-09-14]

ikonar@DT-D80X5M3:~$
```

- Type **gpg --list-keys --keyid-format LONG** and hit enter.

- 

```
ikonar@DT-D80X5M3:~$ gpg --list-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2024-09-13
/home/ikonar/.gnupg/pubring.kbx
------------------------------
pub   rsa3072/F6334B08E1670BDF 2022-09-14 [SC] [expires: 2024-09-13]
      A1E6CC276430053510FC8530F6334B08E1670BDF
uid                 [ultimate] ikonar <indranil.konar@dynatrace.com>
sub   rsa3072/C4E2D0940CAF6415 2022-09-14 [E] [expires: 2024-09-13]

pub   rsa3072/09C60B7345D56F99 2022-09-15 [SC] [expires: 2024-09-14]
      D956BDE6ADD0826F5F97308509C60B7345D56F99
uid                 [ultimate] ikonar <indranil.konar@dynatrace.com>
sub   rsa3072/17240552880E1388 2022-09-15 [E] [expires: 2024-09-14]
```

- Type **gpg a --export 17240552880E1388 > public.key** and hit enter
- Type **gpg a --export-secret-keys 09C60B7345D56F99 > sceret.key** enter **passphase** (abcd for this demo)and hit enter
- 

```
Please enter the passphrase to export the OpenPGP secret key:
"ikonar <indranil.konar@dynatrace.com>"
3072-bit RSA key, ID 09C60B7345D56F99,
created 2022-09-15.


Passphrase: ****_____________________________________________

        <OK>                                    <Cancel>
```

- Type **cat public.key** ,hit enter and copy the key in notepad for use.

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

  mQGNBGMjLR4BDAC8p6CuOxYARkvbPRr5TzoFDujdC/rGoeT6U9pAGCK58VyOxJu+
  DTJwNO7lvXgs7jbXj5z6761ncAfdA8NiohrCZ0QIoBI6wKhLYJ6egoTpt1/4Mt0B
  ****
  -----END PGP PUBLIC KEY BLOCK-----
- Type **cat secret.key** , hit enter and copy the key in notepad for use.

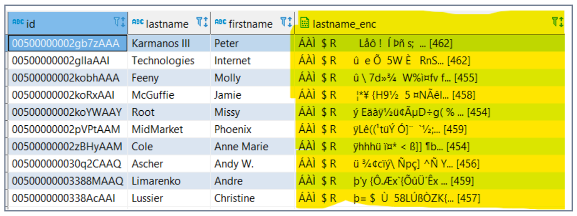
  -----BEGIN PGP PRIVATE KEY BLOCK-----

  lQWGBGMjLR4BDAC8p6CuOxYARkvbPRr5TzoFDujdC/rGoeT6U9pAGCK58VyOxJu+
  DTJwNO7lvXgs7jbXj5z6761ncAfdA8NiohrCZ0QIoBI6wKhLYJ6egoTpt1/4Mt0B
  ****
  -----END PGP PRIVATE KEY BLOCK-----

Step2: Encrypt the data using the public key.

```
SQL1: drop table salesforceprd.user_IK;
SQL2: create table salesforceprd.user_IK( id,lastname ,firstname, lastname_enc) as
select
    user_id_18__c
    , lastname
    , firstname
    , pgp_pub_encrypt(lastname, pubkeys.pubkey) as lastname_enc
from
    salesforceprd."user"
    cross join
    (select dearmor(
'-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGNBGMjLR4BDAC8p6CuOxYARkvbPRr5TzoFDujdC/rGoeT6U9pAGCK58VyOxJu+
DTJwNO7lvXgs7jbXj5z6761ncAfdA8NiohrCZ0QIoBI6wKhLYJ6egoTpt1/4Mt0B
********
-----END PGP PUBLIC KEY BLOCK-----'
) as pubkey) as pubkeys
limit 10
```

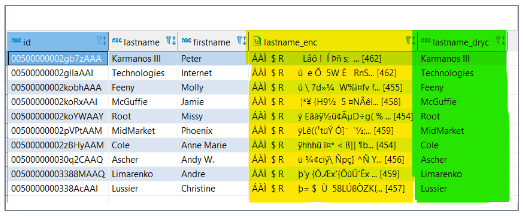
SQL3: select * from salesforceprd.user_IK ;



id	lastname	firstname	lastname_enc
00500000002gb7zAAA	Karmanos III	Peter	ÁÀÌ $ R Låô ! Í Þñ s; ... [462]
00500000002gIlaAAI	Technologies	Internet	ÁÀÌ $ R û e Õ 5W È RnS... [462]
00500000002kobhAAA	Feeny	Molly	ÁÀÌ $ R û \ 7d»¾ W%ì¤fv f... [455]
00500000002koRxAAI	McGuffie	Jamie	ÁÀÌ $ R ¦*¥ {H9½ 5 ¤NÃêl... [458]
00500000002koYWAAY	Root	Missy	ÁÀÌ $ R ý Eäàÿ½ü¢ÃµD÷g(% ... [454]
00500000002pVPtAAM	MidMarket	Phoenix	ÁÀÌ $ R ÿLê((¹tüÝ Ó]¨ `½;... [459]
00500000002zBHyAAM	Cole	Anne Marie	ÁÀÌ $ R ÿhhhü ï¤* < ß]] ¶b... [454]
005000000030q2CAAQ	Ascher	Andy W.	ÁÀÌ $ R ü ¾¢cïÿ\ Ñpç] ^Ñ Y... [456]
0050000003388MAAQ	Limarenko	Andre	ÁÀÌ $ R þ'y {Ô.Æx`{ÕûÜ´Êx ... [459]
005000000338AcAAI	Lussier	Christine	ÁÀÌ $ R þ= $ Ù 58LÚßÒZK{... [457]

Step 3: Decrypt the data using private key and pass phase.

```
select
    id
    , lastname
    , firstname
    ,lastname_enc
    , pgp_pub_decrypt(lastname_enc, pvtkeys.pvtkey,'abcd') as lastname_dryc
from
    salesforceprd.user_IK
    cross join
    (select dearmor(
'-----BEGIN PGP PRIVATE KEY BLOCK-----

lQWGBGMjLR4BDAC8p6CuOxYARkvbPRr5TzoFDujdC/rGoeT6U9pAGCK58VyOxJu+
DTJwNO7lvXgs7jbXj5z6761ncAfdA8NiohrCZ0QIoBI6wKhLYJ6egoTpt1/4Mt0B
*********
-----END PGP PRIVATE KEY BLOCK-----'
) as pvtkey) as pvtkeys
limit 10
```



id	lastname	firstname	lastname_enc	lastname_dryc
00500000002gb7zAAA	Karmanos III	Peter	ÁÀÌ $ R Låô ! Í Þñ s; ... [462]	Karmanos III
00500000002gIIaAAI	Technologies	Internet	ÁÀÌ $ R û e Õ 5W È RnS... [462]	Technologies
00500000002kobhAAA	Feeny	Molly	ÁÀÌ $ R û \ 7d»¾ W%ì¤fv f... [455]	Feeny
00500000002koRxAAI	McGuffie	Jamie	ÁÀÌ $ R ¦*¥ {H9½ 5 ¤NÃêl... [458]	McGuffie
00500000002koYWAAY	Root	Missy	ÁÀÌ $ R ý Eäàÿ½ü¢ÃµD÷g(% ... [454]	Root
00500000002pVPtAAM	MidMarket	Phoenix	ÁÀÌ $ R ÿLê((¹tüÝ Ó]¨ `½;... [459]	MidMarket
00500000002zBHyAAM	Cole	Anne Marie	ÁÀÌ $ R ÿhhhü ï¤* < ß]] ¶b... [454]	Cole
005000000030q2CAAQ	Ascher	Andy W.	ÁÀÌ $ R ü ¾¢cïÿ\ Ñpç] ^Ñ Y... [456]	Ascher
0050000003388MAAQ	Limarenko	Andre	ÁÀÌ $ R þ'y {Ô.Æx`{ÕûÜ´Êx ... [459]	Limarenko
005000000338AcAAI	Lussier	Christine	ÁÀÌ $ R þ= $ Ù 58LÚßÒZK{... [457]	Lussier

Reference

Encrypting data with pgcrypto

Related articles

- How to Analyze and Optimize Postgres
- Encrypt and Decrypt using PGCRYPTO
- Heroku - Create Data Link
- Heroku - CLI Main Commands
- Heroku Prod DB Upgrade