# Integrated Public Number Database (IPND)

## IPND Data Users and Data Providers Access to Internet Interface Service (IIS)

**Date: March 2022**

**Vers: 1.8**

**Approved by: Penny Waite**

**Title: IPND Manager**

**Author(s):** Logical Technologies Pty Ltd

**Application:** Integrated Public Number Database

# 1. TABLE OF CONTENTS

# 2. OVERVIEW

This document describes how to establish a connection to the IPND Internet Interface Service (IIS). It details the technology required.

In order to ensure the confidentiality of the data uploaded and downloaded from the IPND the following measures will be deployed as part of the IIS:

VPN ( SSL ) tunnels

SSH Based file transfer tool with PKI for authentication.

Encryption of files using GnuPG ( open source ) tools also using PKI.

It is assumed that the user has applied and been authorised to become an IPND User by the IPND Manager according to defined processes.
Refer to https://www.telstra.com.au/consumer-advice/ipnd

## 2.1. Assumptions

It has been assumed that Linux users will use command line options and Windows users will use GUIs (although command line options are available)

Filezilla and WinScp have been tested as transfer utilities.

GPG for Windows was used for GPG encryption and decryption.

The solution assumes that the use of OpenVPN will be allowed. In some cases it may be necessary to work with corporate network personnel to allow access.

> **NOTE:** It is the responsibility of Data Users and Data Providers to keep the client utilities up to date.

## 2.2. Information provisioned by LogicalTech

The following table lists the information which will be provided to you for deployment purposes

| Element | Purpose | Section(s) referred |
|---|---|---|
| OpenVPN Username and Password | Access to OpenVPN gateway | 3 VPN |
| Comment details | Input into GPG-key pair | 6 Generating a GPG Key Pair |
| IPND Public Key | Used to encrypt data sent to the IPND | 6 Windows Environment<br>Use one of the integrated GPG management tools such as GNU Privacy Assistant mentioned above.<br>Load the IPND Public key |

## 2.3. Information to be provided to LogicalTech

The following table lists the information which you will send to LogicalTech for deployment purposes. You will need to send details to ipnd-support@logicaltech.com.au.

| Element | Purpose | Section(s) referred |
|---|---|---|
| SSH Public Key | Enable SFTP and SCP access | 4 Authentication |
| GPG Public Key | Enable encryption of files received from the IPND | 6 Generating a GPG Key Pair |

## 2.4. Additional Information

The following table list additional information that will need to be verified. This **must not** be done via email. You will be contacted to verbally verify fingerprints.

| Element | Purpose | Section(s) referred |
|---|---|---|
| Key Fingerprints | Key fingerprints will need to be **verbally** verified. | 12 Appendix 2 – Fingerprints |

# 3. VPN

## 3.1. Overview

In order to reduce the risk profile associated with making sensitive data privately available over the internet access to the IIS will be enabled through TLS VPNs.

This section will describe how to download the configuration files needed to establish a tunnel to the IIS, how to establish a tunnel and how to check that the tunnel has been established.

> **NOTE: The VPN configuration file provided lists the FQDN (Fully Qualified Domain Name) to establish the VPN tunnel.**
> **To ensure high availability of the environment, redundancy has been put in place. This means that in the event of a failure, the VPN services will be started up on another server and assigned another IP address.**
>
> **If your organisation requires firewall restrictions or whitelists to control access, the FQDN must be used rather than a resolved IP address.**

## 3.2. VPN Settings

This information is included within the VPN Configuration file that is downloadable as explained in the following section. Main values are provided as background information. An example of the file is included as APPENDIX 3 – OPENVPN CONFIGURATION FILE EXAMPLE.

| | |
|---|---|
| VPN Gateway URI | gw1.ipnd.com.au |
| VPN Connection Port UDP | 1194 |
| VPN Connection Port TCP | 443 |
| VPN Provisioning URL | https://gw1.ipnd.com.au |

## 3.3. Downloading VPN Configuration

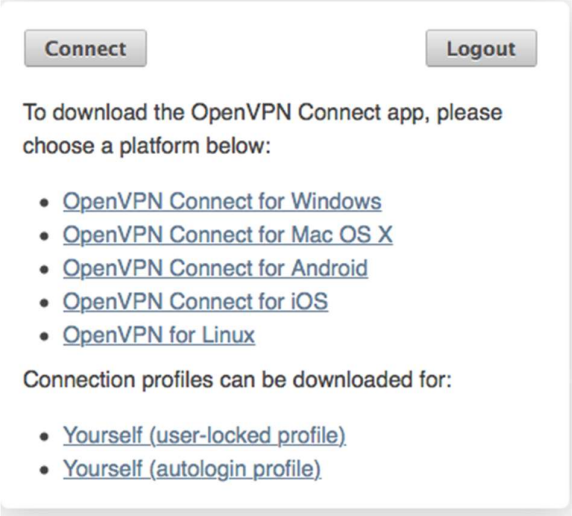
Subsequent to being granted access to the IPND via the IIS the IPND support team will have provided an OpenVPN Username and Password. These will enable you to download the IIS VPN application and configuration files that are available from gw1.ipnd.com.au

Login with the OpenVPN Username and Password provided to you by the IPND Support Team, in the image below the label “user1” is used. Substitute your own credentials for this label.

When connecting initially use the login option as shown below:


Screen 1 - VPN Login

After successfully logging in you will be presented with the following screen:




Screen 2 - VPN Download

The links for downloading the OpenVPN Connect app will direct you to the appropriate download location for the client suited to your environment.

The Windows and OS X links will download a file (.msi or .dmg)

The Android and IOS links will take you the appropriate app stores. The Linux link will take you to additional instructions on how to deploy a Linux distribution OpenVPN client app.

The connection profiles section will enable you to download a client.ovpn configuration file that can be imported into pre deployed SSL clients.

The user-locked profile will require authentication with the password provided. The autologin profile allows the tunnel to be established without a password being entered to establish the tunnel.

## 3.4. Establishing a Tunnel

Invoke the SSL VPN Client as appropriate for your operating system.

An option to connect to gw1.ipnd.com.au will be available. Select that option and establish a connection.

### *LINUX EXAMPLE*

```
sudo openvpn --config client.ovpn
```

### *WINDOWS EXAMPLE*

The Windows Installation of the OpenVPN client will install an icon in the tray or notification area.

To connect:

1. Right Click on the OpenVPN icon and select gw1.ipnd.com.au
2. Then select connect as <user1> (where <user1> is the OpenVPN Username and Password assigned to you by the LogicalTech team)

## 3.6. Checking the Tunnel

The VPN tunnel will have been created in the form of a network interface. For examples see the screen shots below:

### *LINUX EXAMPLE*

```
Ifconfig –a
tun0     Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
         inet addr:10.10.120.6  P-t-P:10.10.120.6  Mask:255.255.255.0
         inet6 addr: fe80::384c:ef9:668b:8cce/64 Scope:Link
         UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:100
         RX bytes:0 (0.0 B)  TX bytes:242 (242.0 B)
```

The above information shows that a virtual networks interface labelled tun0 that has been created by the TLS VPN software. It shows that the local IP address assigned to the interface is 10.10.120.6.

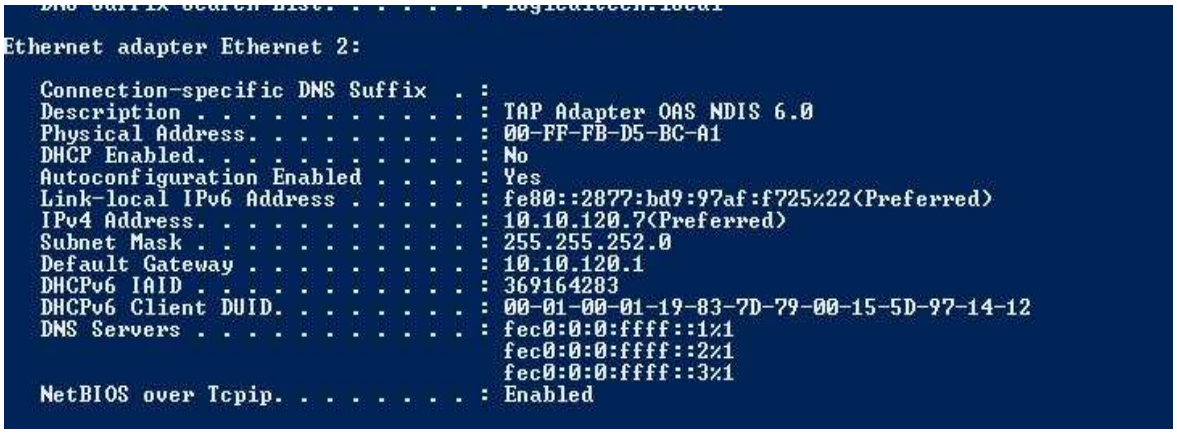
To ensure that data intended for the IIS is routed accordingly a routing table similar to the one displayed in the screen shot below should exist in your system.

```
netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.30.254  0.0.0.0         UG        0 0          0 enp0s25
10.10.110.9     10.10.120.1     255.255.255.255 UGH       0 0          0 tun0
10.10.110.17    10.10.120.1     255.255.255.255 UGH       0 0          0 tun0
10.10.110.18    10.10.120.1     255.255.255.255 UGH       0 0          0 tun0
10.10.110.26    10.10.120.1     255.255.255.255 UGH       0 0          0 tun0
10.10.120.0     0.0.0.0         255.255.255.0   U         0 0          0 tun0
10.10.120.0     10.10.120.1     255.255.254.0   UG        0 0          0 tun0
160.206.232.120 192.168.30.254  255.255.255.255 UGH       0 0          0 enp0s25
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 enp0s25
192.168.30.0    0.0.0.0         255.255.255.0   U         0 0          0 enp0s25
```

### *WINDOWS EXAMPLE*

From a command window run the following command:

```
ipconfig /all
```

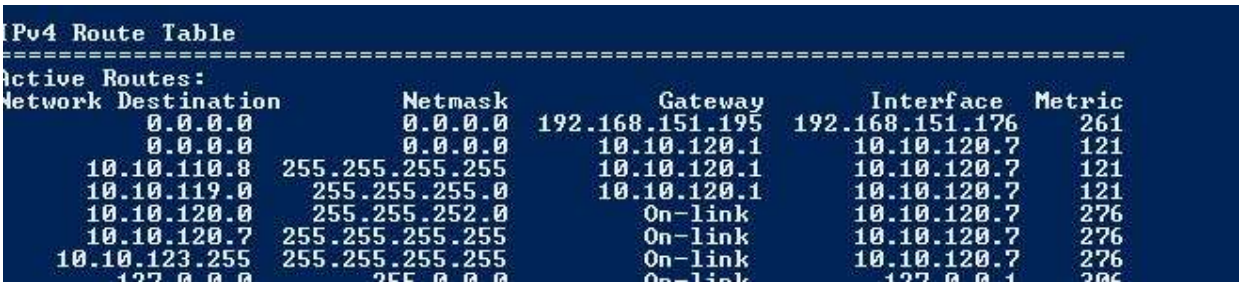



**Screen 3 - VPN Tunnel Windows Example**

The above information shows that a virtual networks interface labelled TAP Adapter has been created by the TLS VPN software. It shows that the local IP address assigned to the interface is 10.10.120.7

Check routing by running the following command from a command window:

```
netstat –rn
```




**Screen 4 - Routing Information - Windows Example**

# 4. AUTHENTICATION

## 4.1. Overview

This section describes the steps required to gain access to the IIS File Transfer Service (FTS) using the SSH based scp and sftp applications. In order to minimise risk associated with password management SSH authentication is based on PKI using SSH key pairs.

### 4.1.1. SSH Key Pairs

SSH Keys are a means of identifying yourself to an SSH server using public-key cryptography.

Public-key cryptography, also known as asymmetric cryptography, is a class of cryptographic algorithms which requires two separate keys, one of which is secret (or private) and one of which is public. Together they are known as a key-pair.

Your Public Key must be provided to LogicalTech once generated. This will be used to register Data User or Data Provider access on the server.

## 4.2. Generating an SSH Key-Pair

#### *LINUX EXAMPLE*

To generate a SSH key pair you will need to use the following command:

```
cd ~/.ssh
ssh-keygen -b 4096 -t ed25519 -f <user> -C <Organisation Name>
```

You will be prompted to provide a secure passphrase.

NOTE: using -t ed25519 implies  a fixed length. If used, the -b flag will be ignored.

> **Note**: that unattended SSH sessions may be initiated without the need to authenticate each connection through the use of the ssh-agent utility. It is preferred that you use this approach. If you do not secure your secret key with a passphrase your access to the IPND may be.

The above command will generate a two key pair files as follows:

```
<user>.pub
<user>
```

-t ed25519 specifies the key type. This is the preferred type of the options that are presented.

> ***Note:*** that key types RSA 1024 bits an DSA are no longer considered secure.

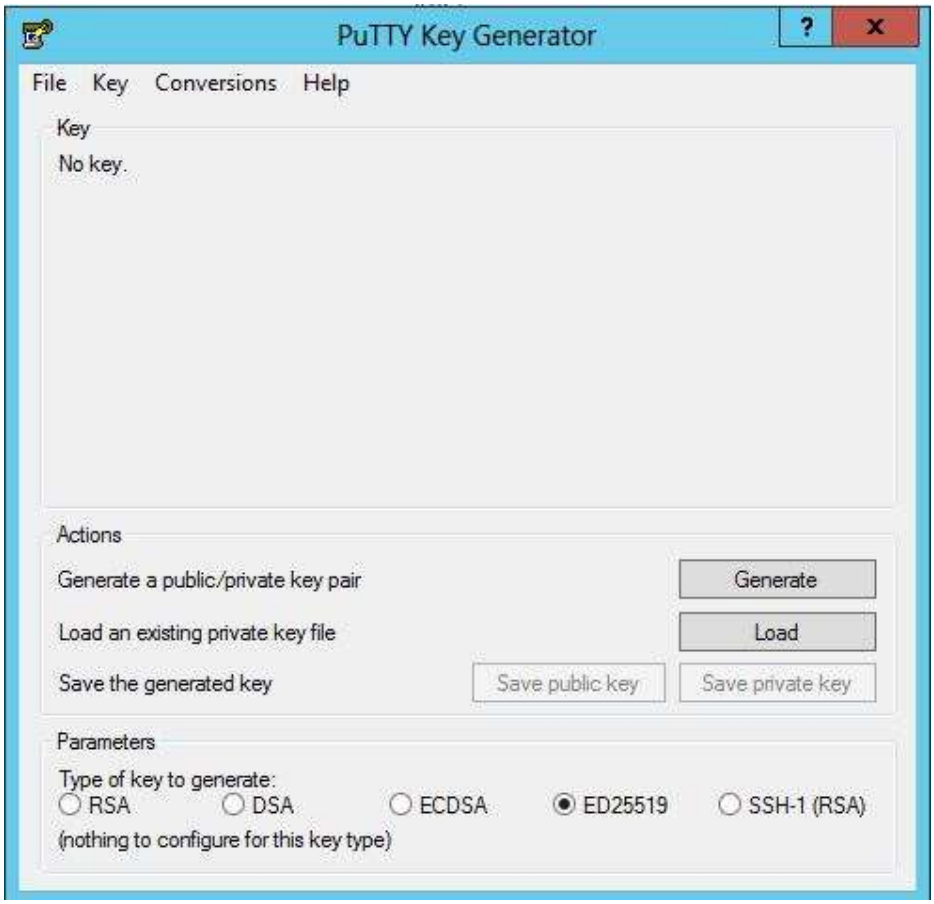
Your private and private keys should be deployed in the <user>/.ssh directory in the user account you may be using.

### *WINDOWS EXAMPLE*

In order to generate an SSH key pair you will need to download a tool such as PuTTygen.

Run the PuTTygen application by clicking on the icon 

You will see a screen similar to this:




**Screen 5 - SSH Key Pair - Windows Example**

You will need to select ED25519 as the key type – then click Generate

Save the public and private keys in a known location.

## 4.3. Public Key

Once you have your keys provide your public key ( <user>.pub ) to ipnd-support@logicaltech.com.au. You will be contacted to validate the key fingerprint before it is deployed.

Refer to Appendix 2 – Fingerprints for details on how to determine key fingerprints.

## 4.4. Using SSH Agent

In order to be able to access the IPND sftp/scp service you will need to authenticate using the private key that you generated in the above step.

This will require that you enter the passphrase associated with your private key. For an unattended batch system this could present a problem and it is therefore advised that the ssh-agent component be used to load the private key so that unattended batch transfers can take place.

ssh-agent acts like a trusted repository into which your private key can be loaded.

### *LINUX EXAMPLE*

Run the program ssh-agent. You may need to do this as root.

This will produce output such as :

```
SSH_AUTH_SOCK=/var/folders/86/7kj3s11j57qgp72g9lcg_rb80000gn/T//ssh-APQFoXMisCuk/agent.53227;
export SSH_AUTH_SOCK;
SSH_AGENT_PID=53228; export SSH_AGENT_PID;
echo Agent pid 53228;
```

> Note - invoking ssh-agent with a "-t nnnn " argument will set the default life time of the loaded key to nnnn. The omission of this parameter ensures that the key is loaded indefinitely.
> This parameter can be overridden with a specific value for a specific key.

This information will provide the ssh-agent details that your process will require to connect to to use the keys. The environment details will need to be made available in any environment that the scp/sftp connections are being used from.

Adding a key requires the invocation of the "ssh-add" command as follows:

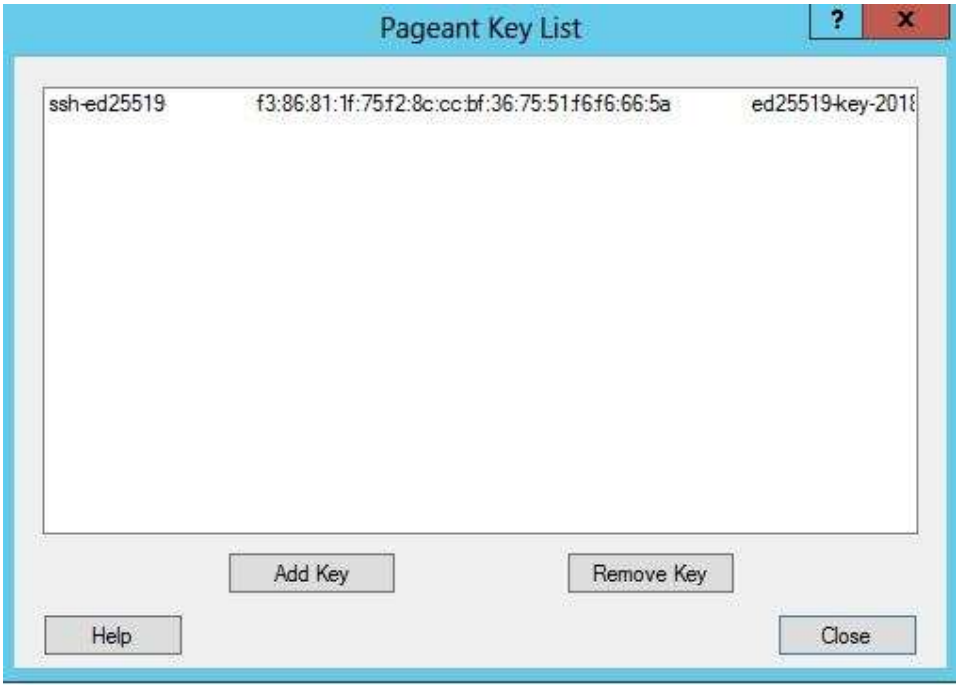
```
ssh-add -t 300 .ssh/<testkey>
```

Will load the private key .ssh/<testkey> into ssh-agent for a duration of 300 seconds. You will be required to enter the passphrase to be able to load the key.

### *WINDOWS EXAMPLE*

You will need to start the putty agent. The Pagent daemon will have been installed with your installation of PuTTyGen.

Start this up (right click) and add your ssh key.




**Screen 6 - Puttygen - SSH Agent**

## 4.5. SSH Access Configuration

### *LINUX EXAMPLE*

In order to ensure that your connection to the IPND services are as seamless as possible ensure that your configuration details are specified in the .ssh/config file associated with the account from which you are transferring files to and from the IPND.

The format of the config file is as follows:

Production Service Example

```
host ipndfts-p
hostname 10.10.110.8
port 22
IdentityFile .ssh/<user_priv_key>
user <prod-user>
```

User Test Service Example

```
host ipndfts-t
hostname 10.10.110.8
port 22
IdentityFile .ssh/<user_priv_key>
user <test-user>
```

### *WINDOWS EXAMPLE*

The configuration files for Windows' GUIs are stored as part of the Site details in the GUI setup.

# 5. CONNECTING

## 5.1. Overview

This section provides some background on, and describes how to connect to the IIS FTS.

## 5.2. VPN

Once you have established a VPN connection as described in the VPN section you can then establish an SSH/SFTP connection. The tunnel needs to be maintained for the duration of the SSH/SFTP session.

## 5.3. Environments

The IIS provides FTS access to the core IPND Production and User Test environments. Refer to Appendix 1 for details of IP addresses etc.

## 5.4. SSH Connection

You will have the option of using two tools to connect to the IPND file transfer service. These are sftp and scp.

NB: Once you have completed file transfers you should terminate BOTH the VPN and SSH connections.

For security reasons VPN and SSH/SFTP sessions are automatically terminated after ~15 minutes of inactivity to the IPND server.

### 5.4.1. SFTP (Linux Examples)

When you invoke sftp you will have a similar interface to standard ftp, such as put, get etc.

> *Note:* SFTP is a better option to use when testing or connecting to the FTS interactively.

### 5.4.2. SCP (Linux Examples)

scp provides an alternative mechanism to send and retrieve files from the IPND FTS services.

> Note SCP is a better option to use in conjunction with automated batch process.

Assuming that the .ssh/config file has been set up as appropriate the syntax for uploading a file using scp as as follows:

```
scp IPNDUPGENTE.0000001.asc ipndfts-p:
```

Where a file is being uploaded to the FTS production environment and the SSH configuration has been setup as described in the SSH Configuration section of this document.

The syntax for downloading a file using scp would be:

```
scp  ipndfts-p:download/IPNDUPGENTE.0000001.nnn.err.asc
```

Where *nnn* represents a retry number. The retry number is used to differentiate each version of the upload file loaded by the Data Provider so that an audit trail is maintained.

To download all err files for a particular upload:

```
scp  ipndfts-p:download/IPNDUPGENTE.0000001.*.err.asc
```

If the .ssh/config file has not been specified then the syntax would be as follows:

```
scp -i .ssh/<user_priv_key>  IPNDUPGENTE.0000001.asc gentelco@ipndfts-p:
```

And for downloading a file would be:

```
scp  i .ssh/<user_priv_key>  gentelco@ipndfts-p:download/IPNDUPGENTE.0000001.nnn.err.asc
```

Where *nnn* represents a retry number. The retry number is used to differentiate each version of the upload file loaded by the Data Provider so that an audit trail is maintained.

To download all err files for a particular upload:

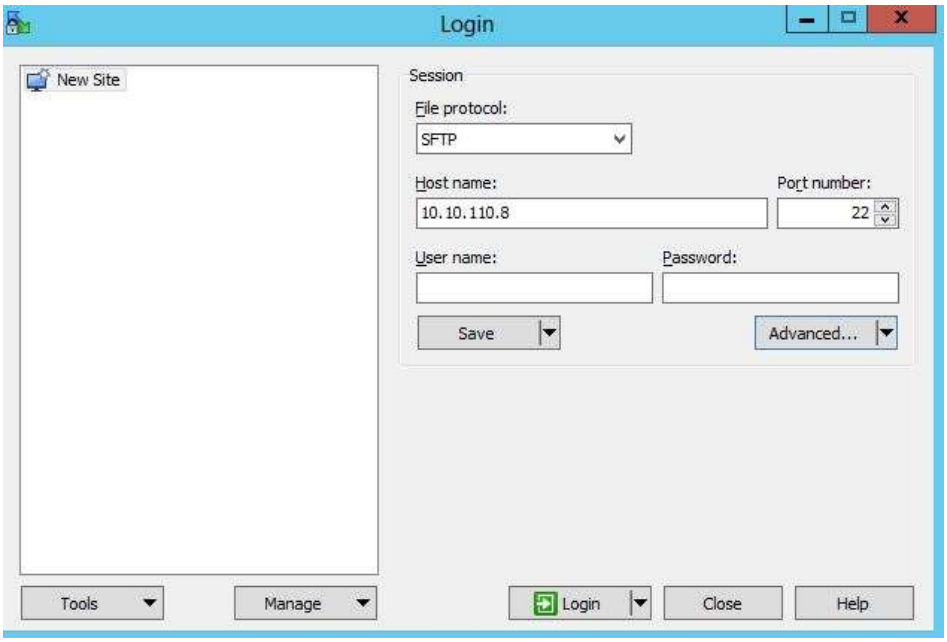
```
scp  i .ssh/<user_priv_key>  gentelco@ipndfts-p:download/IPNDUPGENTE.0000001.*.err.asc
```

### 5.4.3. SCP and SFTP (Windows Environment)

You will need to install software that will allow SCP or SFTP connection.

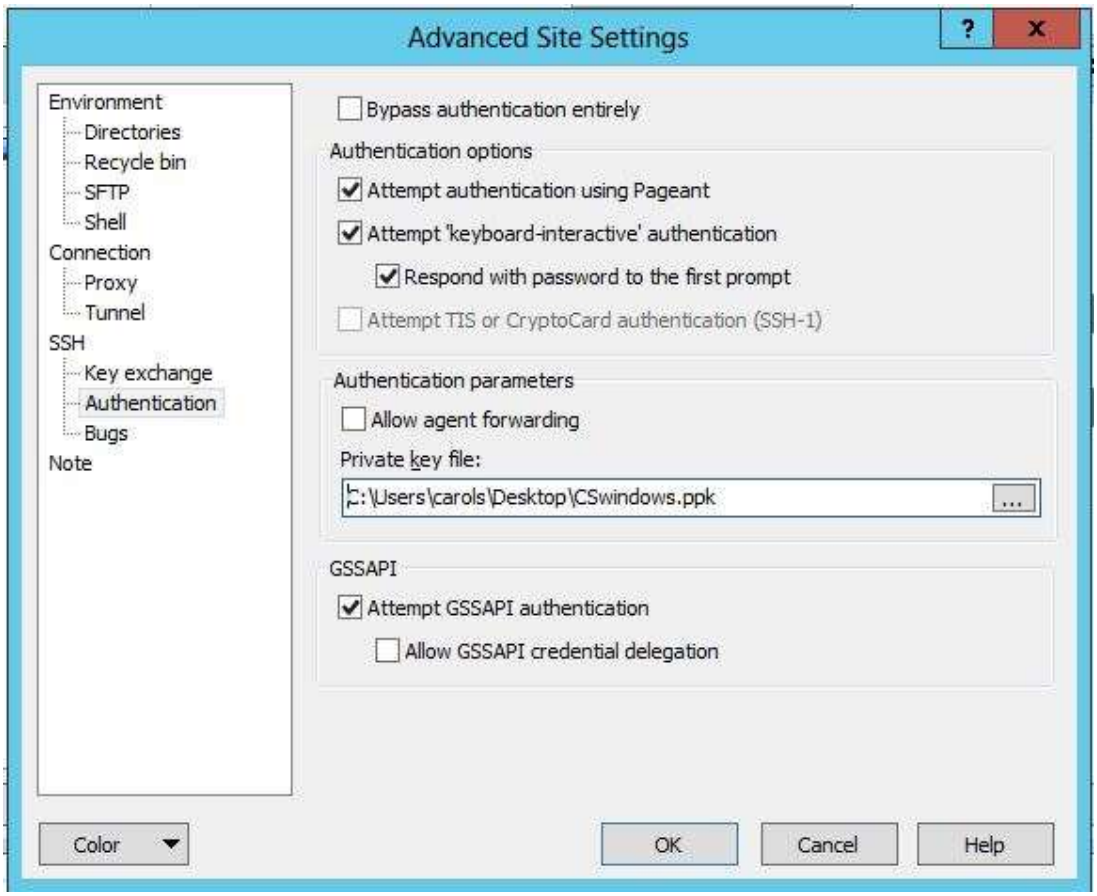
WinSCP or Filezilla are example clients.

#### *WINSCP CONFIGURATION*




**Screen 7 - WinSCP Configuration**

Select the Advanced button



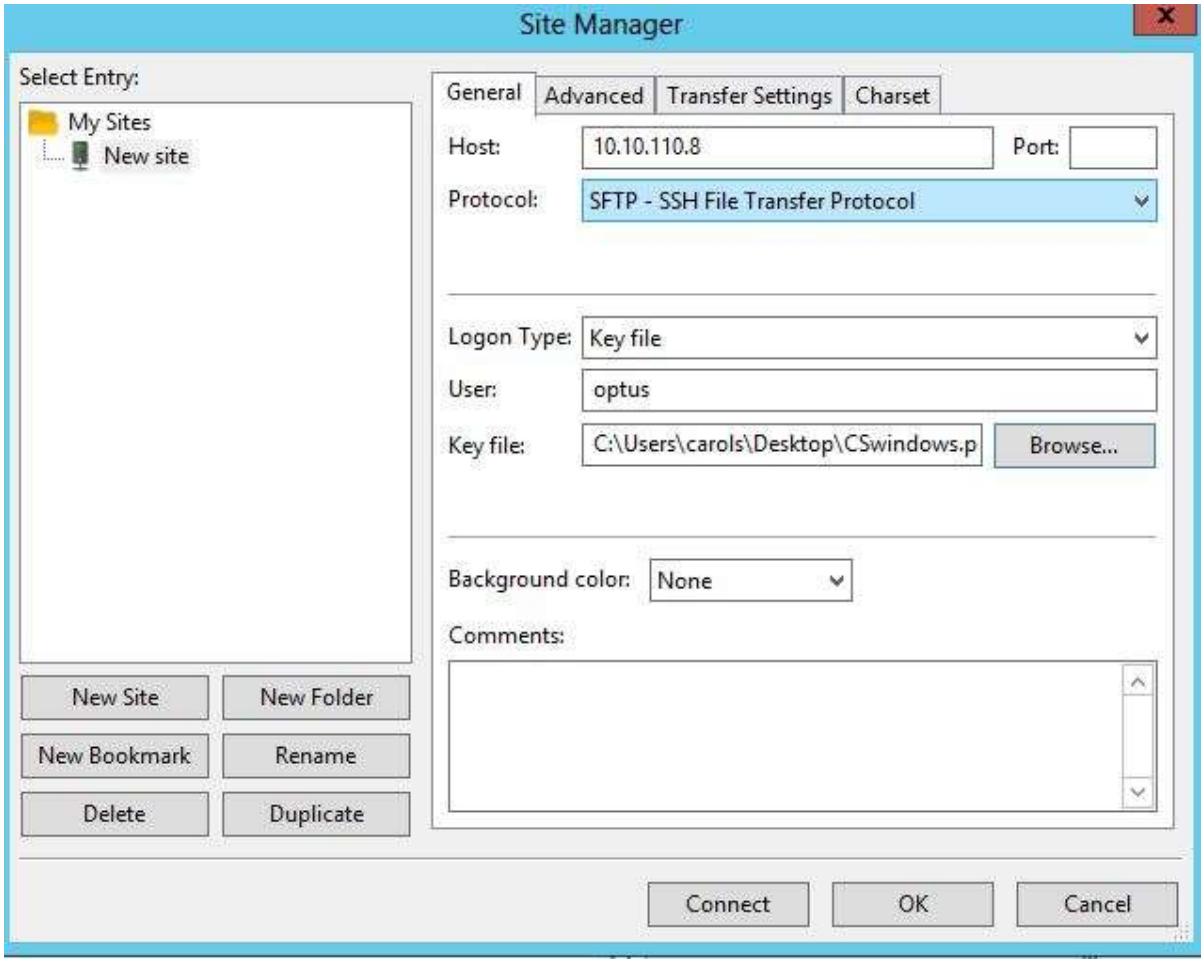

**Screen 8 - WinSCP - Site Settings**

In the SSH Authentication section, select the private key you generated previously.

You can save the login for your organisation in the User name: input box.

Save the configuration as a name you will use again.

Click Login to log into the server.




**Screen 9 - Filezilla Configuration**

Create a new site.

Specify the IP Address

Specify the username provided for your organisation

Select SFTP as the Protocol

Select Key File in Logon Type.

In the Key File box, browse to the file that contains your SSH private key and add it.

## 5.5. Directories

| |
|---|
| NOTE. Relative path names must be used rather than absolute paths.<br><br>e.g. download/ rather than /data/prod/home/p_<user>/download/ |

Each FTS environment consists of the user's home directory and four subdirectories. Files must only be uploaded to the home directory. Processed files and files available for download can be found in the subdirectories which are:

1. download
2. received

3. rejected
4. archived

**download**

This directory will contain non-archived IPND files that you may download.

**received**

This directory will contain a copy of uploaded IPND files that pass initial validation. The file names are prefixed with a unique timestamp code. It is these files that are sent to the IPND system for processing.

NB: The files will appear in this directory ~ 5 seconds after being received.

| Note: files must be only be uploaded into the user's home directory, not any other subdirectory. |
|---|

**rejected**

This directory will contain zero byte (i.e empty) versions of any uploaded IPND files that fail initial validation or are unable to be decrypted on the legacy system

The files are prefixed with a unique timestamp code and a suffix indicating the reason for failure. Refer to section 9 Messages for a full list of the suffixes and failure reasons. Examples of failure reasons are:

1. the filename does not comply with the specified valid filename associated with the user account,
2. the file has not been encrypted.

**archived**

This directory will contain further subdirectories for archived download, received and rejected IPND files. It will contain files that are older than 6 months but still within the archive retention period.

# 6. File Encryption

## 6.1. Overview

All files that are provided to and from the IPND via the IIS FTS will be encrypted using GnuPG

This section provides an overview on how to use the programs and utilities associated with this software.

## 6.2. GnuPG Key Pairs

GnuPG uses public-key cryptography so that users may communicate securely. In a public-key system, each user has a pair of keys consisting of a private key and a public key. A user's private key is kept secret; it need never be revealed. The public key may be given to anyone with whom the user wants to communicate. GnuPG uses a somewhat more sophisticated scheme in which a user has a primary keypair and then zero or more additional subordinate keypairs. The primary and subordinate keypairs are bundled to facilitate key management and the bundle can often be considered simply as one keypair

## 6.3. Generating a GPG Key Pair

This section describes how to generate a GPG key pair so that IPND files can be encrypted and decrypted.

> Note: these should be created in the same environment in which IPND files are going to be sent/received from the IPND.

Your Public Key must be provided to LogicalTech once generated. This will be used to encrypt files you receive from the IPND. Please send the key to IPND-support@logicaltech.com.au.

### *LINUX EXAMPLE*

To generate a GPG key pair run the following command on the CLI.

```
GPG --full-generate-key
This will result in the following be displayed:
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
```

Select 1 as shown above. You will then be asked about keysize.

```
What keysize do you want? (2048) 4096
```

Select 2048 or 4096. You will then be asked about the lifetime or duration of the key.

```
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)
```

Select 0 to ensure that you do not need to renew the key.

You will then be asked to input your name, email address and Comment and to confirm that the information is ok.

```
Real name: Test User
Email address: test-user@org.com.au
Comment: Org
```

You will be provided with the information that needs to be added to the Comment field.

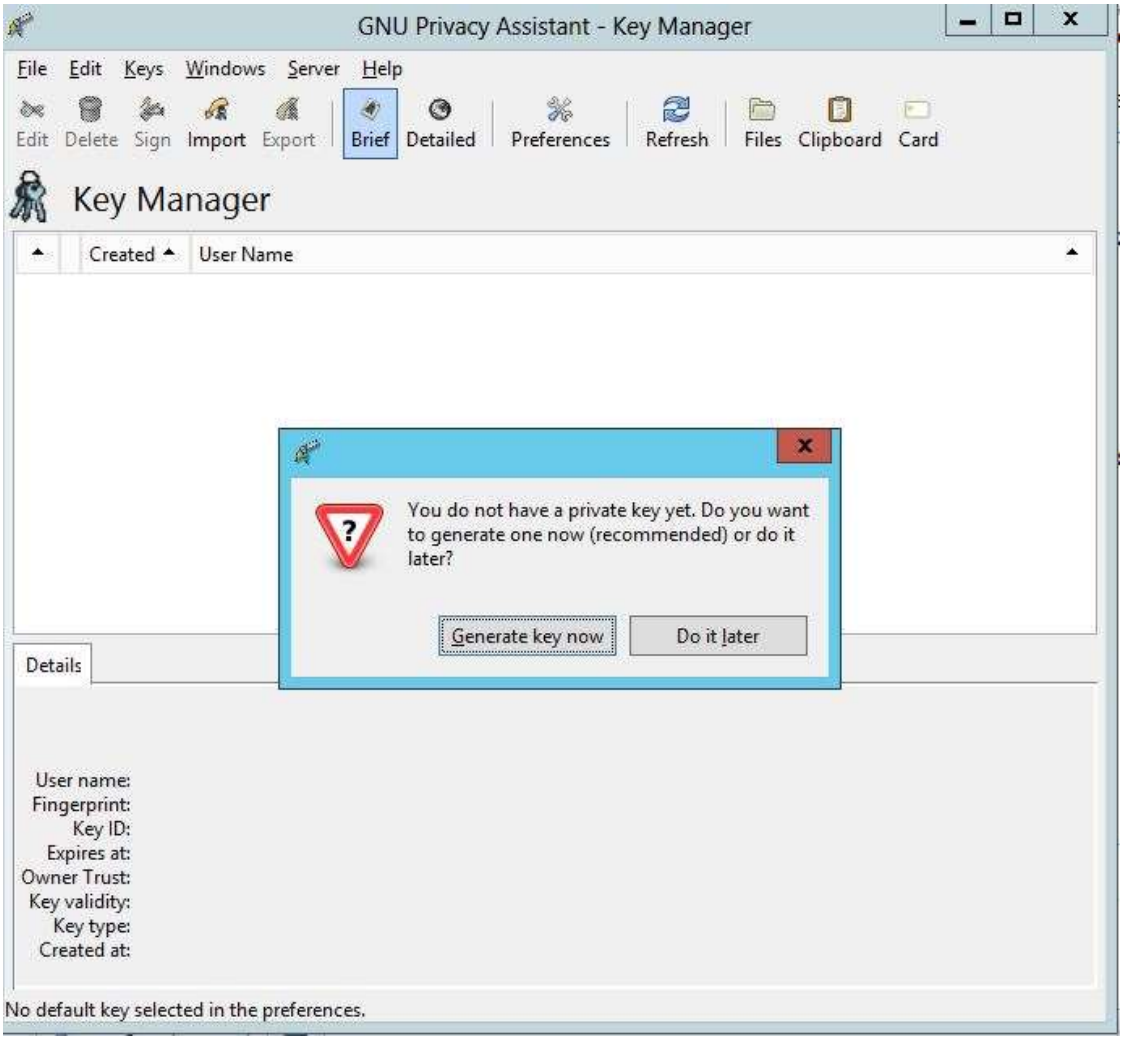
You will be requested to input a passphrase to protect your key - it is imperative that you specify a strong password or passsphrase. After your passphrase has been typed in the keys will be created and stored in your key chain.

To check that your keys are loaded run the following command and check that the keys are displayed.

```
GPG —list-keys
```

### *WINDOWS EXAMPLE*

You will need to install software that will allow you to generate GPG keys. You can review options at https://www.gnupg.org/download/




**Screen 10 - GNU Privacy Assistant - Key Generation**

You will be asked to input your name and email address.

You must enter the Comment provided by LogicalTech in the Comment field

You will also have the option to generate a passphrase.

Save a backup of the key.

## 6.4. Managing your GPG key ring

There are 3 main keys that you will need to mange:

1. Your secret key generated in the Generating a GPG Key Pair section outlined above.
2. Your public key also generated in the Generating a GPG Key Pair section outlined above ( together 1 and 2 constitute a key-pair)
3. The IPND public key.

Your key pair will be stored in a GPG key ring.

### *Linux Environment*

The keys created by you will have been added to your key ring when you created the key pair. You can verify this by running one of the following commands.

```
gpg —list-keys
gpg —list-secret-keys
gpg —list-public-keys
```

### *Windows Environment*

Use one of the integrated GPG management tools such as GNU Privacy Assistant mentioned above.

## 6.5. Load the IPND Public key

This key will have been provided to you by the IPND Support Team. To import into your key ring run the following command ( it assumes the IPND Public key is named IPND-IIS-public.key )

### *Linux Environment*

```
gpg —import IPND-IIS-public.key
```

Verify the fingerprint of the IPND IIS Public key.

```
gpg --fingerprint ipnd-iis@ipnd.com.au
pub   2048R/5BC59835 2018-01-23
    Key fingerprint = CD4C 04C9 630D AD81 B192  A1BC 460F 8D7C 5BC5 9835
uid      [ultimate] ipnd-iis (IPND IIS file transfer encryption key.) <ipnd-iis@ipnd.com.au>
sub   2048R/FBFA9B6E 2018-01-23
```

Confirm with the LogicalTech team that the fingerprint is correct.

Refer to Appendix 2 – Fingerprints for details on how to determine key fingerprints.

**NOTE: Please do not send key fingerprints via email. They must be verified verbally.**

Set the trust of the IPND IIS Public key.

```
gpg —edit-key ipnd-iis@ipnd.com.au
```

Type in “trust” in the CLI that is invoked and then select option 5 and then exit.

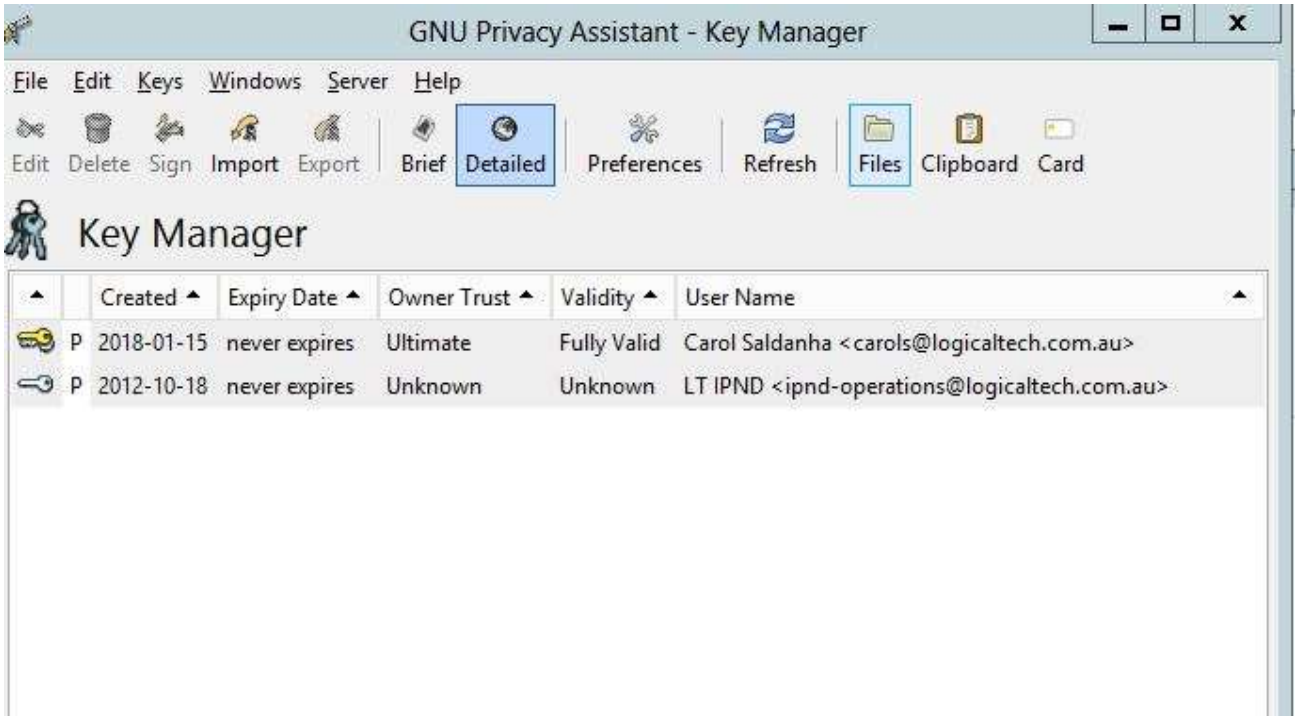
```
trust
  1 = I don't know or won't say
  2 = I do NOT trust
  3 = I trust marginally
  4 = I trust fully
  5 = I trust ultimately
  m = back to the main menu
```

### *WINDOWS ENVIRONMENT*

You will need to import the IPND Public Key provided to you by the IPND Support Team.

You can verify that the keys are loaded by using the GPA application gui.

To do this click on keyring in the main menu. The following screen will be displayed.




**Screen 11 - Importing Keys into Keyring**

You will need to "trust" the IPND Public key. All files you receive will be "signed" with the IPND private key to prove it was generated by the IPND.

Trusting the IPND public key will allow the file to be unencrypted.

Right click on the IPND key and click on "Set Owner Trust". Set the trust level to "Ultimate".

## 6.6. Encrypting and Decrypting Files

### *LINUX ENVIRONMENT*

**Encrypting**

Files that you upload to the IPND IIS will need to be encrypted using the IPND public key.

The command is:

```
gpg —batch —sign —encrypt --armor --recipient ipnd-iis@ipnd.com.au IPNDUPGENTE.0000001
```

This will produce a file named IPNDUPGENTE.0000001.asc which will have been encrypted with the IPND IIS public key. The —sign option will also have signed the file with your private key.

| Note: All files uploaded to the IPND IIS will need to be encrypted as described here. |
|---|

**Decrypting**

All files that you download from the IPND IIS will have been encrypted using YOUR public key and will need to be decrypted using YOUR secret key.

To decrypt a file the command is:

```
gpg —batch —decrypt IPNDUPGENTE.0000001.001.err.asc
```

After running this command the file will be decrypted using your secret key and a plain text file called IPNDUPGENTE.0000001.001.err will have been created.

### *WINDOWS ENVIRONMENT*

**Encrypting**

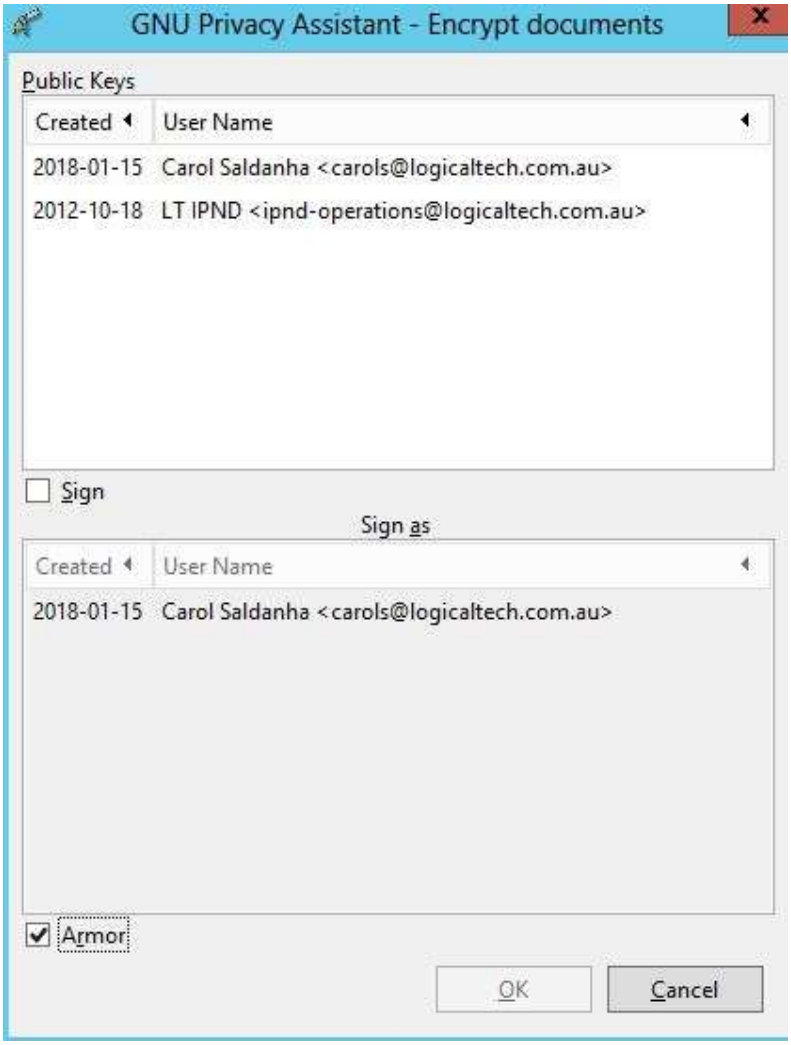
Click on the Files menu option in the GNU Privacy Assistant GUI.

Select the file you want to encrypt e.g IPNDUPGENTE.0000001

Select the Encrypt menu option.

You will be presented with the options of which public keys to use to encrypt the file. Select the IPND public key.

You must also check the “armor” checkbox




**Screen 12 - Encrypting Files**

The resulting file will be called IPNDUPGENTE.0000001.asc

> NOTE: the IPND will expect to receive an encrypted file with a “asc” extension. You must therefore select the “armor” option.

**Decrypting**

Click on the Files menu option in the GNU Privacy Assistant GUI.

Select the file you want to decrypt e.g IPNDUPGENTE.0000001.001.err.asc

Select the decrypt menu option.

Your private key should be used to decrypt the file you have downloaded.

# 7. BATCH PROCESSING

This section provides a brief overview of how to setup your environment to use GPG in a batch mode. This is an involved area and this information is provided as an initial aid.

Depending on your operating environment your secret key will be cached up by a GPG agent and you should only need to specify your secret key passphrase the first time.

It is possible to cache your secret key for a specified duration so that any batch processing can occur without requiring user input.

# 8. File Names

## 8.1. Data Providers

### 8.1.1. Upload File

#### *Customer Record System Extract IPND Upload File*

Data provider upload file formats are defined in section 6.1.2 of the Data Users and Data Providers Technical Requirement for the IPND document.
Data Provider upload files must be encrypted (refer 6.6 Encrypting and Decrypting Files ) with the IPND public key and the filename must be in the format:

IPNDUP<XXXXX>.<NNNNNN>.asc

Where

<XXXXX> is a valid File Source for the Data Provider ,

<NNNNNNN> is the file sequence number with a leading 0

“asc” indicates that the file has been encrypted using the documented method.

> Note the “.asc” extension is created by the gnupg tools that will encrypt the data for transmission. Refer to the File Encryption section for the data encryption process. Files that have not been encrypted will not be accepted by the FTS.

Filenames that do not match the expected format will be rejected. The Filesource <XXXXX> must be one associated with the ssh account. This will have been configured when the account is set up.

Example:

An IPND Data Provider with an (ssh) account of p_gentelco has been assigned a filesource of GENTE.

Using this ssh account, tt will only be possible to upload a file in the following format:

IPNDUPGENTE.NNNNNNN.asc

An invalid pattern will result in an empty file in the rejected directory. This empty file will have the same invalid filename, be prefixed with a timestamp and suffixed with R001.

### 8.1.2. Download Files

Download files will be encrypted with the public gpg key provided by the individual organisation. Refer to Section 6.6 Encrypting and Decrypting Files.

#### *IPND error file Report to Data Providers*

For each upload file received from a data provider that passes initial reject processing, there will be a corresponding error file. The error file format is described in section 6.1.3 of the Data Users and Data Providers Technical Requirement for the IPND document.

The error file name will be based on the name of the upload file for which it is generated. Each upload file is given a retry number when it is received by the IPND. This retry number is used to differentiate each version of the upload file loaded by an IPND Data Provider for audit trail purposes.

The filename format of the error file in the IIS is described below:

IPNDUP<XXXXX>.<NNNNNN>.<MMM>.err.asc

Where

| | |
|---|---|
| <XXXXX> | is the File Source for the Data Provider |
| <NNNNNNN> | is the sequence number of the associated upload file |
| <MMM> | is the retry number i.e indicates the number of times sequence <NNNNNNN> has been uploaded |
| "err" | indicates an error file associated with sequence <NNNNNNN> and retry <MMM> |
| "asc" | is the suffix indicating that the file is encrypted using the gpg public key associated with the Data Provider |

**NOTE**: All error files include a retry suffix in the IIS environment. There is no symbolic link to the latest .err file as in the IPND Legacy environment.

### *DATA PROVIDER SNAPSHOT FILE*

These reports are produced on request from a Data Provider. The output format for these files once decrypted will be as described in Section 6.1.20 of the Data Users and Data Providers Technical Requirement for the IPND document.
The filename format of these reports in the IIS is as follows:

IPNDRU.<XXXXX>.<MMMMMMM>.<NNNNNNN>.<PPP(P)>.asc

Where

| | |
|---|---|
| <XXXXX> | Is the File Source code for the Data Provider |
| <MMMMMMM> | is the run number identifying the output file. It indicates that the file was part of the MMMMMMM$^{th}$ run of an extract for this filesource. |
| <NNNNNNN> | is the file sequence number of the last upload file from the Data Provider with this File Source. |
| <PPPP> or <PPP> - | Span number. All Output On Request download files are limited to 100,000 rows of data. The first 100,000 rows are written to span 001. Requests that generate > 100,000 rows are written to additional spanned files. If the number of output files exceeds 999, the spanned number will increment to 1000 onwards. |
| | If the number of output files exceeds 999, the spanned number will increment to 1000 onwards. |
| "asc" | is the suffix indicating that the file is encrypted using the gpg public key associated with the Data Provider |

#### *CHANGED DATA PROVIDER REPORT*

The Changed Data Provider Report runs monthly on 1st of the month and documents all Data Provider changes from the previous month. The format of the file once decrypted is described in Section 6.1.19 of the Data Users and Data Providers Technical Requirement for the IPND document.

The filename format in the IIS is described below:

IPNDDP<XXXXX>.<NNNNNNN>.asc

Where

| | |
|---|---|
| <XXXXX> | is the File Source code for the Data Provider |
| <NNNNNNN> | is the File sequence number with leading 0 |
| "asc" | is the suffix indicating that the file is encrypted using the gpg public key associated with the Data Provider |

#### *DATA PROVIDER QUERY FILE*

The DPQF is produced as part of the processing from the Data User Query File Sub-system. This file is described in detail in section 5.2.5.3.1 of the Data Users and Data Providers Technical Requirements for the IPND.

The format of this file once decrypted is described in Section 6.1.17 of the same document.

The filename format in the IIS is described below:

IPNDQP.<XXXXX>.<NNNNNNN>.asc

Where

| | |
|---|---|
| <XXXXX> | is the File Source for the Data Provider |
| <NNNNNNN> | is a file sequence number with leading 0 |
| "asc" | suffix indicating that the file is encrypted using the gpg public key associated with the Data Provider. |

## 8.2. Data Users

### 8.2.1. Upload File

#### *IPND DATA USER QUERY FILE DUQF*

Data Users can upload a Data User Query File. The DUQF Sub-system is described in Section 5.2.5 of the Data Users and Data Providers Technical Requirements for the IPND.

The format of this file is described in Section 6.1.13 of the same document.

The filename format in the IIS is described below:

IPNDQU<XXXXX>.<yyyymmddhhmmss>.asc

Where

| | |
|---|---|
| <XXXXX> | is a valid File Source for the Data User, |
| <yyyymmddhhmmss> | is a date stamp indicating when the file was created. |

“asc” indicates that the file has been encrypted using the documented method.

> Note the “.asc” extension is created by the gnupg tools that will encrypt the data for transmission. Refer to the File Encryption section for the data encryption process. Files that have not been encrypted will not be accepted by the FTS.

Filenames that do not match the expected format will be rejected. The Filesource <XXXXX> must be one associated with the ssh account. This will have been configured when the account is set up.

Example:

An IPND Data User with an (ssh) account of p_genuser has been assigned a filesource of GUSER.

Using this ssh account, it will only be possible to upload a file in the following format:

IPNDQUGUSER.yyyymmddhhmiss.asc

An invalid pattern will result in an empty file in the rejected directory. This empty file will have the same invalid filename, be prefixed with a timestamp and suffixed with R001.

### 8.2.2. Download Files

#### *IPND Download Files to Data Users*

The IPND will produce a distinct Download File for each Data User. There are 6 types of Data Users who receive Download Files. For more information refer to Section 5.2.2.7.2 of the Data Users and Data Providers Technical Requirements for the IPND. File formats for these files are described in Section 6.1.5 to 6.1.11 of the Data Users and Data Providers Technical Requirements for the IPND.

The filename format for these files in the IIS is described below:

IPND<TT>.<XXXXX>.<NNNNNNN>. asc

where

<TT> refers to the file type and may be one of

"ES" for Emergency Services
"LA" for Law Enforcement Agencies
"DI" for Directory Publishers and Directory Assistance
“LD” for Location Dependent Carriage Service
“RS” for Researcher
“EW” for Early Warning System

<XXXXX> refers to the individual DU File source code
<NNNNNNN> refers to a sequence number uniquely enumerating the output file

“asc” is the suffix indicating that the file is encrypted using the gpg associated with the data user

### *OUTPUT ON REQUEST EXTRACT FILES*

Data User output on request files are generated on a request basis. Refer section 5.2.6 to of the Data Users and Data Providers Technical Requirements for the IPND document for more detail. The file format of these files is described in Section 6.1.5 to 6.1.11 of the Data Users and Data Providers Technical Requirements for the IPND.

The filename format of output on request files for data users in the IIS is described below;

IPND<TT>.<XXXXX>.<MMMMMMM>.<NNNNNNN>.<PPP(P)>

Where

| | |
|---|---|
| "IPND" | literal Identifier String |
| <TT> | file Type – may be one of the following:<br>"RE" for Emergency Services<br>"RL" for Law Enforcement Agencies<br>"RI" for Directory Publishers<br>"RD" for Location Dependent Carriage Service providers<br>"RR" for Researchers<br>"PR" for Health and Public Policy Researchers |
| <XXXXX> | refers to the DU file source code |
| <MMMMMMM> | run number uniquely identifying the output file. It indicates that a file was part of the MMMMMMM$^{th}$ run for that download file type <TT> |
| <NNNNNNN> | file sequence number with leading 0 |
| <PPPP> or <PPP> - | Span number. All Output On Request download files are limited to 100,000 rows of data. The first 100,000 rows are written to span 001. Requests that generate > 100,000 rows are written to additional spanned files. If the number of output files exceeds 999, the spanned number will increment to 1000 onwards. |
| "asc" | suffix indicating the file is encrypted with the gpg public key associated with the data user. |

### *IPND DUQF ERROR FILE*

The DUQF Error file is produced as part of the DUQF sub-system described in 5.2.5 of Data Users and Data Providers Technical Requirements for the IPND. The format of these files is described in Section 6.1.14 of the same document.

DUQF err files are produced for every DUQF upload.

The filename format in the IIS is as follows:

IPNDQU<XXXXX>.<yyyymmddhhmmss>.<MMM>.err.asc

Where

| | |
|---|---|
| <XXXXX> | is the DU file source code |
| <yyyymmddhhmmss> | is a date time stamp indicating when the file was created |
| <MMM> | is an optional retry number (in case a duplicate DUQF filename was received) |
| "err" | indicates an error (status ) file |

| | |
|---|---|
| “asc” | suffix indicating that the file is encrypted using the gpg public key associated with the data user |

### *AMALGAMATED QUERY FILE PROCESS (DAQF)*

The DAQF file is produced monthly as part of the DUQF sub-system. Refer to section 5.2.5.4 of the of Data Users and Data Providers Technical Requirements for the IPND for details and Section 6.1.18 of the same document for the format.

The filename format of the DAQF in the IIS is as follows:

IPND<TT>.<XXXXX>.<NNNNNNN>.asc

Where

| | |
|---|---|
| <TT> | refers to the file type and may be one of<br>"QE" for a Emergency Services DAQF<br>"QL" for a Law Enforcement Agencies DAQF<br>“QW” for a Early Warning System DAQF<br>"QI" for a Directory Publishers DAQF<br>“QD” for a Location Dependent Carriage Service DAQF<br>“QS” for a Researcher DAQF |
| <XXXXX> | is the DU file source |
| <NNNNNNN> | is a file sequence number with leading 0 |
| “asc” | suffix indicating that the file is encrypted using the gpg public key associated with the Data User |

# 9. MESSAGES

Error Files will report success or failure of uploads as documented in Section 6.1.4 of the Data Users and Data Providers Technical Requirements for the IPND document.

**NOTE**: All error files include a retry suffix in the IIS environment. There is no symbolic link to the latest .err file as in the IPND Legacy environment.

Files that are immediately rejected i.e. not sent to the legacy IPND system will be moved to the "rejected" directory as an empty file with an error suffix. The error suffix will represent the first reason found for rejecting the file.

Files that are not able to be decrypted will result in:

- an empty file in the "rejected" directory with a R020 suffix
- a copy of the original file in the "received" directory for auditing purposes

The table below documents rejected file errors:

| Error | Reason |
|---|---|
| R001 | File does not match valid file pattern |
| R002 | File not valid for user type |
| R003 | For IPNDUP/QU files, the FILESOURCE muse be valid for user |
| R010 | File must be encrypted |
| R020 | File decryption failed |
| R100 | Empty File |
| R200 | File too large. This is an upper extreme limit applied to the encrypted file. It is separate to the row count applied by the legacy system. |

# 10. REFERENCES

For detailed information on how to use the gnupg set of tools refer to:

https://www.gnupg.org/documentation/manuals/gnupg.pdf

## 10.1. Glossary

| Term | Description |
|---|---|
| CLI | Command Line Interface |
| FTS | File Transfer Service |
| GPG | GNU Privacy Guard |
| IIS | Internet Interface Service |
| IPND | Integrated Public Number database |
| PKI | Public Key Infrastructure |
| SSH | Secure Shell |
| TLS | Transport Layer Security |
| VPN | Virtual Private Network |

# 11. APPENDIX 1

This page provides a summary of all the relevant information required.

Note: Until 04/02/2022 there will be two possible IP addresses available for the FTS Usertest environment. Post cutover to the new environment only one IP address will be used as documented below.

| | |
|---|---|
| FTS Production IP address | 10.10.110.8 |
| Before 04/02/2022<br>FTS User Test IP Address | 10.10.110.8<br>or<br>10.11.110.8 |
| After 04/02/2022<br>FTS User Test IP Address | 10.11.110.8 |
| FTS Production Username | “p_<user>” |
| FTS User Test Username | “t_<user>” |

# 12. Appendix 2 – Fingerprints

## 12.1. SSH

The following section describes how to obtain fingerprints for ssh and gpg keys.

**Never send key fingerprints via email. You will be contacted to verify key fingerprints verbally.**

### *Linux*

```
ssh-keygen -lf /path/to/ssh/key
```

or to see an MD5 hash of it if generated using putty

```
ssh-keygen -E md5 -l -f ./t_genba.pub
```

Converting Windows Key to Unix

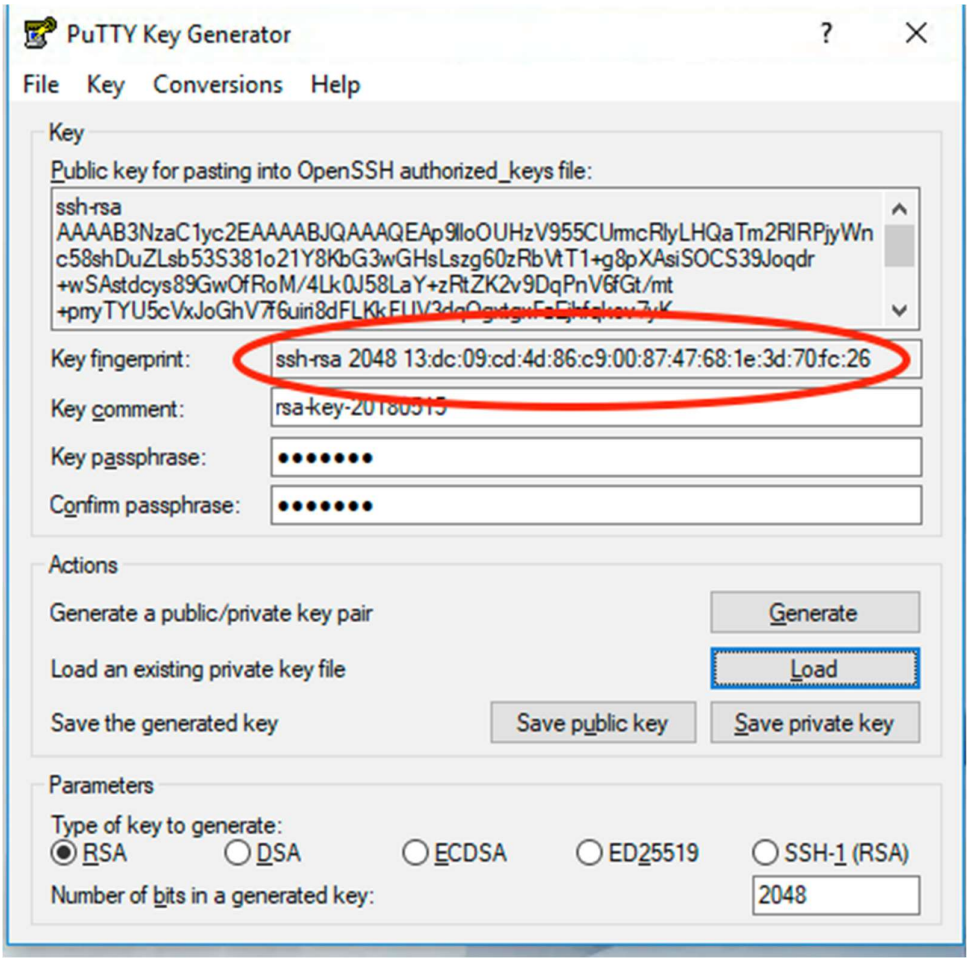
```
ssh-keygen -i -f keyfile.pub > newkeyfile.pub
```

### *Windows*

When generating using putty key gen look in the Key fingerprint: field

```
ssh-rsa 2048 13:dc:09:cd:4d:86:c9:00:87:47:68:1e:3d:70:fc:26
```

Alternatively load the key and read the fingerprint (you will need to provide the pass phrase to load the key.




Screen 13 - Fingerprint Identification

## 12.2. GPG

### *Linux*

Once key has been loaded into keychain:

```
gpg --fingerprint <user>
pub   rsa2048 2013-06-11 [SC]
    3777 04C9 34DD 3DAB DBED  D500 8947 60EF 77FD D883
uid         [  full  ] User Name <user@email.com>
sub   rsa2048 2013-06-11 [E]
```

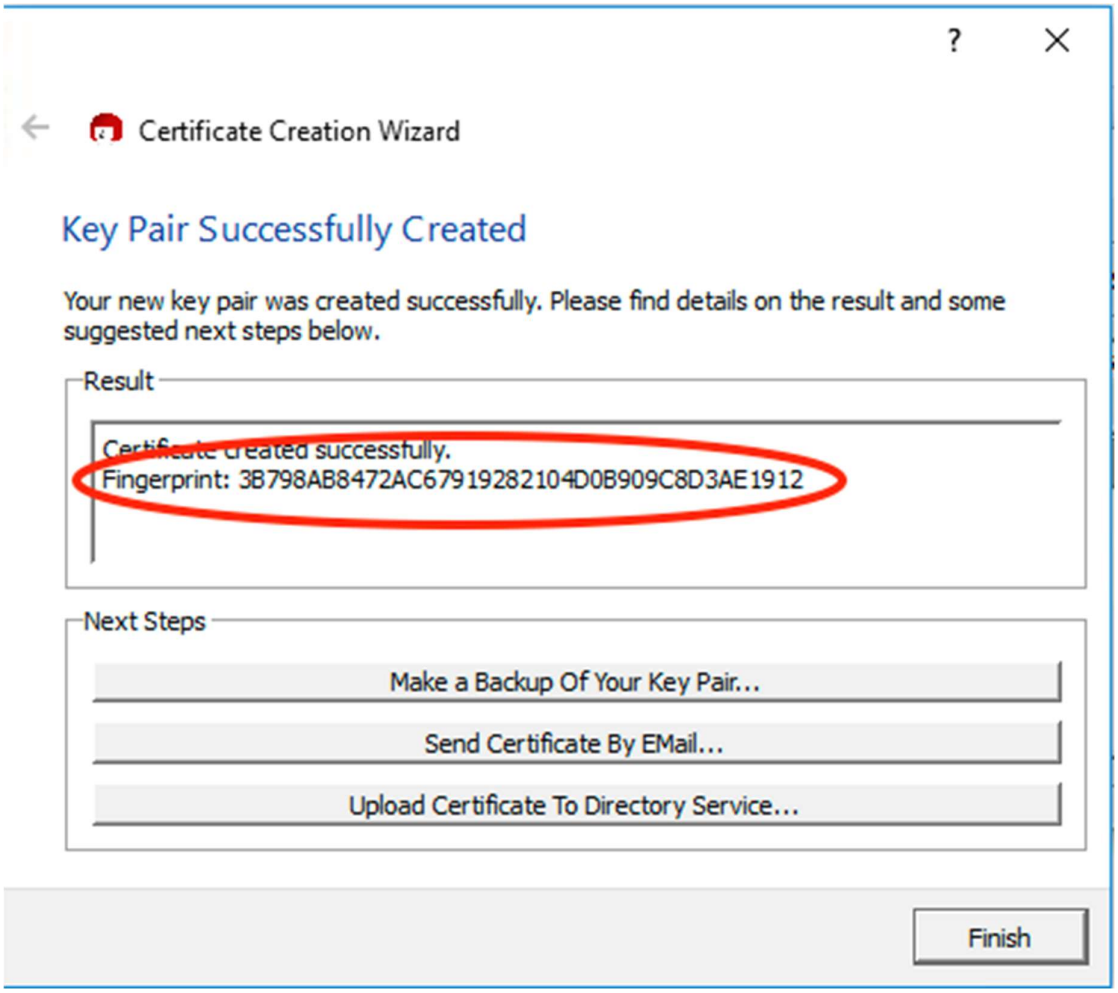
Checking file if not loaded:

```
gpg <key>.asc
gpg: WARNING: no command supplied.  Trying to guess what you mean ...
pub   rsa2048 2013-06-11 [SC]
    377704C934DD3DABDBEDD500894760EF77FDD883
uid         User Name <user@email.com>
sub   rsa2048 2013-06-11 [E]
```

### *Windows*

**Kleopatra**

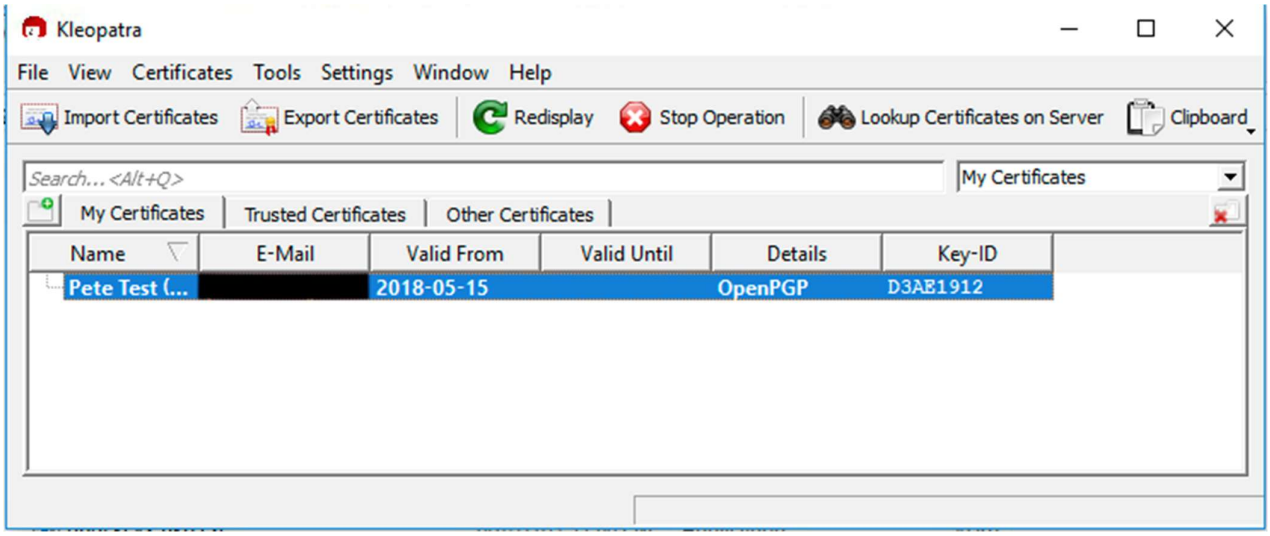
When gpg key pair is created.



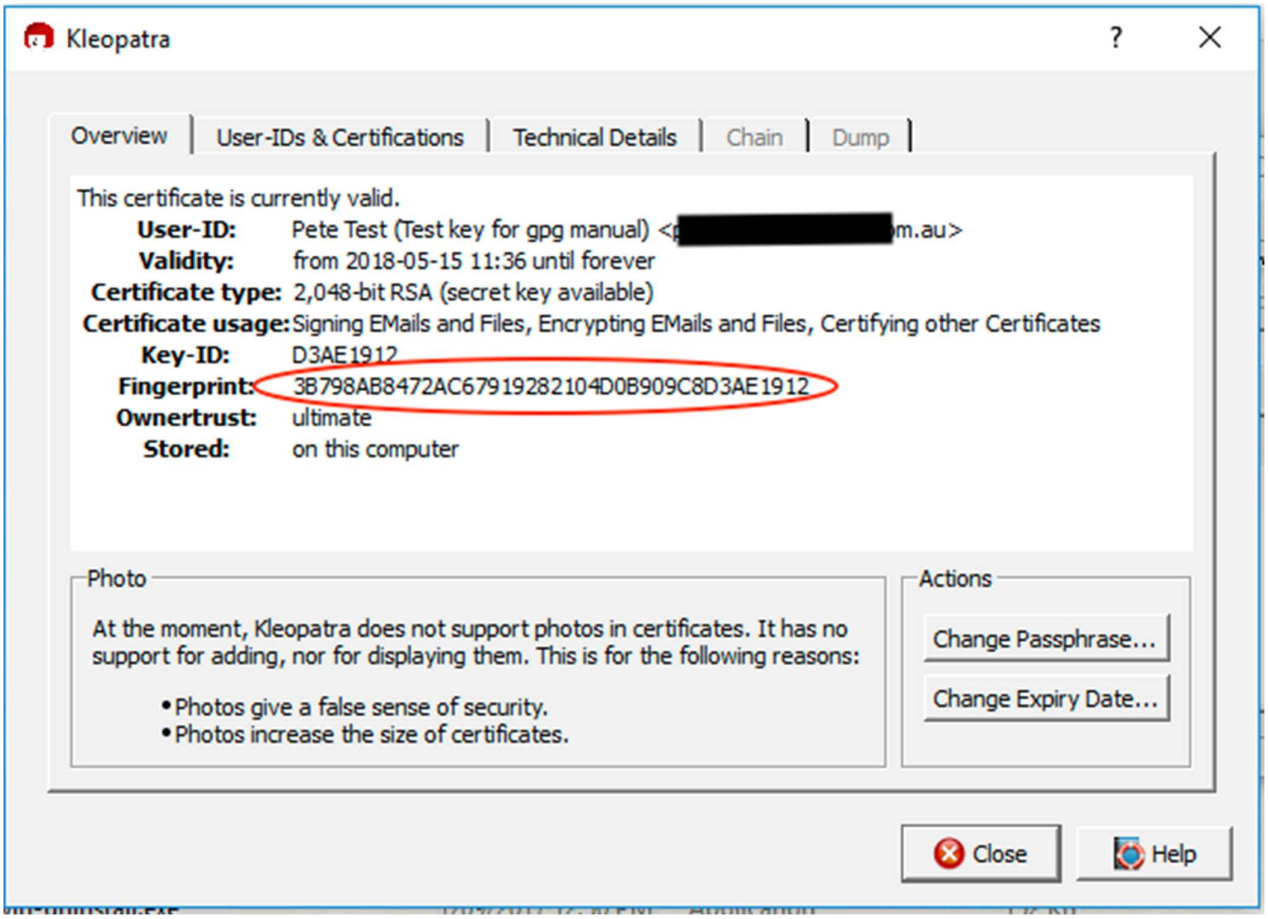

Screen 14 - GPG Fingerprint - Kleopatra

After key pair has been generated:

Double click on the key pair.




Screen 15 - GPG Fingerprint - Kleopatra - post create




Screen 16 - GPG Fingerprint - Kleopatra - post create 2

# 13. APPENDIX 3 – OPENVPN CONFIGURATION FILE EXAMPLE

```
# Automatically generated OpenVPN client config file
# Generated on Thu May  3 09:30:38 2018 by pvpn01
# Note: this config file contains inline private keys
#       and therefore should be kept confidential!
# Note: this configuration is user-locked to the username below
# OVPN_ACCESS_SERVER_USERNAME=<user>
# Define the profile name of this particular configuration file
# OVPN_ACCESS_SERVER_PROFILE=<user>@gw1.ipnd.com.au/AUTOLOGIN
# OVPN_ACCESS_SERVER_AUTOLOGIN=1
# OVPN_ACCESS_SERVER_CLI_PREF_ALLOW_WEB_IMPORT=True
# OVPN_ACCESS_SERVER_CLI_PREF_BASIC_CLIENT=False
# OVPN_ACCESS_SERVER_CLI_PREF_ENABLE_CONNECT=True
# OVPN_ACCESS_SERVER_CLI_PREF_ENABLE_XD_PROXY=True
# OVPN_ACCESS_SERVER_WSHOST=gw1.ipnd.com.au:443
# OVPN_ACCESS_SERVER_WEB_CA_BUNDLE_START
# -----BEGIN CERTIFICATE-----

# <information removed>

# -----END CERTIFICATE-----
# -----BEGIN CERTIFICATE-----


# -----END CERTIFICATE-----
# -----BEGIN CERTIFICATE-----
# <information removed>
# -----END CERTIFICATE-----
# OVPN_ACCESS_SERVER_WEB_CA_BUNDLE_STOP
# OVPN_ACCESS_SERVER_IS_OPENVPN_WEB_CA=0
# OVPN_ACCESS_SERVER_ORGANIZATION=Telstra IPND IIS
setenv FORWARD_COMPATIBLE 1
client
server-poll-timeout 4
nobind

remote gw1.ipnd.com.au 1194 udp
remote gw1.ipnd.com.au 443 tcp

dev tun
dev-type tun
ns-cert-type server
setenv opt tls-version-min 1.0 or-highest
reneg-sec 604800
sndbuf 100000
rcvbuf 100000
# NOTE: LZO commands are pushed by the Access Server at connect time.
# NOTE: The below line doesn't disable LZO.
comp-lzo no
verb 3
setenv PUSH_PEER_INFO

<ca>
-----BEGIN CERTIFICATE-----
```

```
<information removed>

-----END CERTIFICATE-----
</ca>

<cert>
-----BEGIN CERTIFICATE-----

<information removed>

-----END CERTIFICATE-----
</cert>

<key>
-----BEGIN PRIVATE KEY-----
<information removed>
-----END PRIVATE KEY-----
</key>

key-direction 1
<tls-auth>
#
# 2048 bit OpenVPN static key (Server Agent)
#
-----BEGIN OpenVPN Static key V1-----
<information removed>
-----END OpenVPN Static key V1-----
</tls-auth>

## -----BEGIN RSA SIGNATURE-----

<information removed>

## -----END RSA SIGNATURE-----
## -----BEGIN CERTIFICATE-----

<information removed>

## -----END CERTIFICATE-----
## -----BEGIN CERTIFICATE-----

<information removed>


## -----END CERTIFICATE-----
## -----BEGIN CERTIFICATE-----

<information removed>

## -----END CERTIFICATE-----
## -----BEGIN CERTIFICATE-----

<information removed>

## -----END CERTIFICATE-----
```