For Regular Retroshare Nodes optionally routed via Tor and/or I2P

*** Important Note *** Even when a Retroshare Regular Node optionally routes their datastreams additionally through the Tor and I2P networks to connect to a Retroshare Hidden Node routed via Tor or I2P, the Retroshare Regular Nodes Ipv4 address is still shown in the details of your connection to the remote friend running the Retroshare Hidden Node routed via Tor or I2P. If you do not want even your direct friends to see your Ipv4 address, you'll want to generate a new Retroshare Hidden Node using the advanced option and route your new Retroshare Hidden Node via Tor or I2P. Those only display your xxx.onion or xxx.i2p encoded, encrypted routing network address to your directly added friends which is pretty secure.

*** Update *** Retroshare Developers are now updating the Retroshare Source code to remove the display of the peers IPv4 Address in the Network Peer--Details connection information from a Regular Retroshare Node connecting to a Hidden Retroshare Node via Tor and/or I2P as that IPv4 Address doesn't serve to connect to the directly added peers Retroshare Hidden Node and wouldn't otherwise show anyway in the remote friends firewall connection. Only localhost, 127.0.0.1 addresses are actually used in this configuration to connect to the Retroshare Hidden Nodes tor xyz.onion address inside the Tor Network itself.

Retroshare Regular Nodes optionally routing via Tor or I2P to connect to Retroshare Hidden Nodes operating via Tor or I2P

To optionally/additionally route your Retroshare Regular Node traffic via Tor, install the Tor binary whose default port is 9050. Run/Start Tor as a user, not as root/superuser. Then Only After starting Tor and Tor creates the new network , bootup your Retroshare Regular Node and confirm your Outgoing Connections are additionally being routed via Tor.

https://www.torproject.org/download/download.html.en <-Windows Tor
https://www.torproject.org/download/download-unix.html.en <-Linux/Unix
Debian,Ubuntu,Knoppix, CentOS, Fedora, Gentoo, FreeBSD, OpenBSD, NetBSD,
Same Link Source Tarballs ./configure && make && src/or/tor
https://www.torproject.org/docs/debian.html.en <-Repository Steps
https://www.torproject.org/docs/rpms.html.en <--Install Tor Guides

If you wish to build the latest tor binary from the source code follow this Linux box terminal mode example.

To build the Tor source binary don't use ./configure && make && src/or/tor
Instead break this into separate stepped commands
$ ./configure
$ make
$ sudo make install

Configure the torrc file if you need to add a custom national exit node
I prefer to change ownership of the torrc file from root to local user, however I leave that up to you. Otherwise you'll need root/superuser access in order to edit the torrc file. The standard torrc (tor configuration) file that comes with tor is typically either packed with optional settings for hidden services, bridges and transparent proxys (the full monte') or really dumb-ed down and barely useable. For a fully operational torrc file example that has multiple useful options commented out, refer to the Hidden Retroshare Node routed via Tor-I2P document where I show a fully functional torrc file series of top-bottom commands in its entirety.

```
ExitNodes {us}
StrictNodes 1
SocksPort 9050

Then start tor

$ tor
```

Read the tor notices, warnings as your new Tor Router install creates a new tor circuit at 100%. If there's any problems they'll be reported.

The main overlying point here is when you start-up the tor binary, carefully read any notices and warnings that are posted as Tor creates the 100% new networked tor circuit. If Tor cannot locate the expected torrc file or its command set is incorrect, Tor often throws a warning and invents a useless torrc dumbed down series of commands. If that happens then stop/kill tor and correct the problem with a working, proper torrc file in the precise directory path Tor is expecting and posted in the notices.

In Retroshare, select Options--Network--Hidden Service Configuration tab

```
Outgoing Connections
Tor Socks Proxy 127.0.0.1 9050  Green Icon if working, Black Icon if not
```

A Green Icon indicates you are good to go with your Retroshare Regular Node routing optionally via Tor. You can now additionally add Retroshare Hidden Node routed via Tor friends to your keyring and connect to their Retroshare Hidden Node.

==========

Retroshare Regular node I2P Client Tunnel creation and setup

For Retroshare 0.6 Regular Nodes wanting to connect to a peers Retroshare I2P routed Hidden Node (eepsite), you would first download and install the I2P Router.

https://geti2p.net/en/download

```
$ i2prouter start <-- Run, Startup the Garlic I2P Router

Your default web browser should automatically load-up if not already running and
show http://127.0.0.1:7657/home  The I2P ROUTER CONSOLE in your browser.
click on the 'LOCAL TUNNELS' button
You should now be at http://127.0.0.1:7657/i2ptunnelmgr
HIDDEN SERVICES MANAGER
click on the 'Tunnel Wizard' button
Client Tunnel enabled, select 'next'
Tunnel type, select SOCKS 4/4a/5 in drop down menu, then select 'next'
Enter your Client Tunnel Name and Description
Example TAS_I2P_Client, Client I2P RS06_TAS_I2P_tunnel, then select 'next'
Leave 'Outproxy' blank and select 'next'
Binding Address and Port
Example 127.0.0.1 8555 and then select 'next'
Tunnel auto-start enabled, then select 'Finish'
Double-check and review the Wizard Completed Summary on your choices and copy down
the settings and Port number if needed, then select 'Save Tunnel'.

The summary of your new I2P Client Tunnel Settings
```

Working Example
Wizard completed (creating the I2P Client)
.
The wizard has now collected enough information to create your tunnel. Upon clicking the Save button below, the wizard will set up the tunnel, and take you back to the main I2PTunnel page. Because you chose to automatically start the tunnel when the router starts, you don't have to do anything further. The router will start the tunnel once it has been set up.
.
Below is a summary of the options you chose: Server or client tunnel? Client
Tunnel type SOCKS 4/4a/5
Tunnel name and description TAS_I2P_Client
Client I2P RS06_TAS_I2P_tunnel
Tunnel destination
Binding address and port
8555
127.0.0.1
Tunnel auto-start Yes

Alongside these basic settings, there are a number of advanced options for tunnel configuration. The wizard will set reasonably sensible default values for these, but you can view and/or edit these by clicking on the tunnel's name in the main I2PTunnel page.
Bootup Retroshare06 if it isn't already running
Select Options - Network - Hidden Service Configuration - Outgoing Connections

I2P Socks Proxy
Using the above example I'd enter 127.0.0.1 8555
You can try localhost instead of 127.0.0.1 whichever works.
Then confirm the icon next to I2P outgoing Okay is 'Green'.

This Finishes Up the Regular Retroshare node Client I2p configuration which then allows you to peer/friend Hidden node Retroshare Server Eepsites.

==========