Tor

Introducción

Tor (https://www.torproject.org) es una "intrared" dentro de Internet que implementa una técnica de enrutamiento diferente de las técnicas estándares llamada Onion Routing, gracias a la cual se garantiza el **anonimato** de los extremos y la **privacidad** de los datos transferidos. Tor es una red operada por voluntarios que enmascara quién eres y desde dónde estás conectado. También te protege en la misma red Tor, ya que puedes estar seguro que permanecerás anonimo frente a otros usuarios de Tor también.

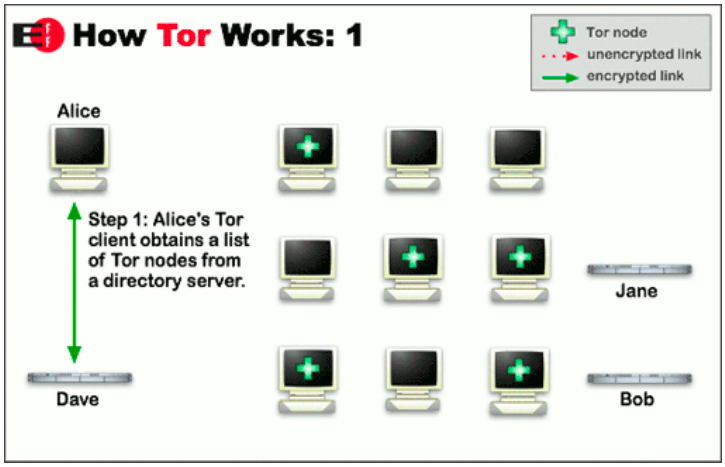
El enrutado tradicional que usamos para conectarnos a servidores en Internet es directo: de tu ordenador a tu router, de ahí a los enrutadores de tu ISP (proveedor de Internet) y después directos al servidor de destino elegido. Fácil y sencillo, salvo por el hecho de que si alguien intercepta los paquetes de datos en un punto intermedio sabrá perfectamente de dónde vienen y a dónde van. Incluso aunque se cifren los datos de cada paquete (por ejemplo, visitando una página HTTPS) las cabeceras de este no se cifran, y los campos del remitente y destinatario (entre otros) siguen siendo visibles. Ahí es donde entra el Onion Routing. Expliquémoslo.

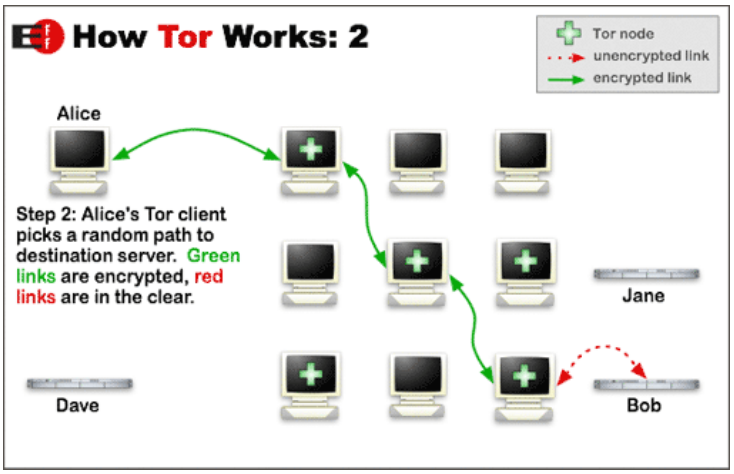
Primero, el ordenador A, que quiere enviar el mensaje a B, calcula una ruta más o menos aleatoria al destino pasando por varios nodos intermedios. Después, consigue las claves públicas de todos ellos usando un directorio de nodos. Usando cifrado asimétrico, el ordenador A cifra el mensaje como una cebolla: por capas: primero cifrará el mensaje con la clave pública del último nodo de la ruta, para que sólo él lo pueda descifrar. Además del mensaje, incluye (también cifradas) instrucciones para llegar al destino, B. Todo este paquete, junto con las instrucciones para llegar al último nodo de la lista, se cifra de nuevo para que sólo lo pueda descifrar el penúltimo nodo de la ruta. El proceso se repite hasta que acabamos con todos los nodos de la ruta. Con esto ya tenemos el paquete de datos listo, así que toca enviarlo. El ordenador A conecta con el primer nodo de la ruta, y le envía el paquete. Este nodo lo descifra, y sigue las instrucciones que ha descifrado para enviar el resto del paquete al nodo siguiente. Éste descifrará de nuevo y volverá a enviar al siguiente, y así sucesivamente. Los datos llegarán finalmente al nodo de salida, que enviará el mensaje a su destino.

Ninguno de los nodos, salvo el primero y el último, saben de dónde viene o a dónde va el mensaje. Ni siquiera saben qué posición ocupan en la ruta, y mucho menos conocen el contenido del mensaje. De esta forma, aunque se intercepten las comunicaciones entre dos nodos, es imposible saber qué datos transmite, a dónde van o de dónde vienen. Incluso aunque hubiese un nodo infiltrado, un topo en la red, no tendría nada que hacer con los mensajes que recibe. También tiene la ventaja de que es muy difícil tumbar la red Tor: al estar los nodos distribuidos, habría que tumbar todos y cada uno de ellos para poder parar las comunicaciones.

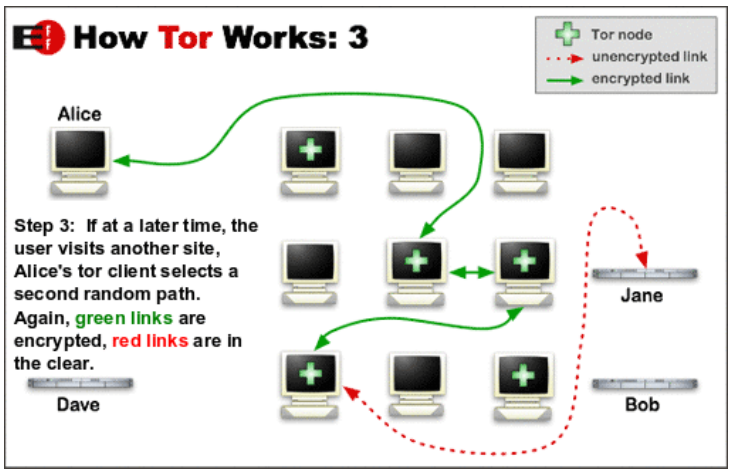
Por otro lado, también hay que tener en cuenta que el nodo final de salida puede leer el mensaje original, así que también hay que cifrar el mensaje original.

Todos estos pasos están ilustrados en las siguientes imágenes:





NOTA: The last connection would be encrypted if Alice were visiting an HTTPS website: keep in mind that Tor cannot encrypt your traffic after it leaves the Tor network.



Para el lector interesado, en https://securityinabox.org/en/guide/anonymity-and-circumvention se desarrollan diversos conceptosrelacionados con la esquivación de la **censura** en Internet, un aspecto donde la red Tor también puede ayudar.

Note there is a trade-off between anonymity and speed: bouncing your traffic through several servers in various parts of the world will almost always be slower than a direct connection to the Internet.

NOTA: Para poder establecer los circuitos virtuales en la red Tor, es necesario disponer de una lista que detalle el estado (dirección, ancho de banda, etc.) de los distintos nodos disponibles. Este documento es llamado "Consenso". En cada cliente Tor se encuentra embebido la información de 10 nodos confiables que son la base de la red Tor, con IPs conocidas. La misión de estos nodos es de actualizar y mantener el Consenso, por lo cual son llamados Autoridades de directorio (Directory authorities). El proceso se resume como:

Cada autoridad genera una lista de nodos.
Cada autoridad calcula los estados de los nodos, el ancho de banda que disponen y distintos flags.
Una vez calculados estos parámetros, la autoridad sube esta información (voto) para el resto de autor.
Cada autoridad descarga los votos del resto, combina la información, recalcula y firma el resultado.
Se sube nuevamente esta información firmada al resto de autoridades.
Si hay mayoría, se valida el consenso y es publicado por cada autoridad.
El proceso anterior y la actualización del consenso se llevan a cabo cada una hora.

Una lectura interesante sobre esto es https://blog.torproject.org/lifecycle-new-relay

The Tor protocol is detailed here: https://gitweb.torproject.org/torspec.git/tree/ . Other very complete websites explaining inners of Tor network are https://github.com/tfukui95/tor-experiment and https://witestlab.poly.edu/blog/anonymous-routing-of-network-traffic-using-tor/

Tor Browser

Para quienes necesitan ocasionalmente de anonimato y privacidad cuando navegan por sitios web, el "Tor Browser" ofrece una manera rápida y fácil de utilizar la red Tor. Este navegador (https://www.torproject.org/projects/torbrowser.html.en) no es más que una versión modificada del navegador Firefox (por lo tanto, libre) que es capaz de:

*Ocultar la IP de los usuarios
*Eliminar cualquier tipo de sistema de seguimiento online
*Habilitar el acceso a páginas web prohibidas (o solo existentes dentro de la propia red Tor)

As we already know, the Tor network consists of thousands of servers run by volunteers all over the world. Every time the Tor Browser makes a new connection, it selects three of these Tor relays and connects to the Internet through them. When you use the Tor Browser, your internet traffic will appear to come from a different IP address (often in a different country); this different IP belongs to the last "frontier" Tor node, responsible to "go out" from Tor to Internet. As a result, the Tor Browser hides your IP address from the websites you access while also hiding the websites you access from third parties who might try to monitor your traffic. It also ensures that no single Tor relay can figure out both your location on the Internet and the websites you visit (though some of them will know one or the other).

Además, el "Tor Browser" no requiere instalación: basta con descomprimir el paquete zip obtenido tras su instalación y hacer doble clic sobre el ejecutable. Por tanto, es una solución portable.

Before you extract the Tor Browser package, you should verify that it is authentic doing this:

0.-Right-click the *(sig)* link just beneath the Tor Browser's download button and save the resulting file (it has the ".asc" extension). You will need this file to verify the authenticity of the Tor Browser package.

1.Import the Tor Project's public signing key (0x4E2C6E8793298290) by executing following command:

gpg --keyserver x-hkp://pool.sks-keyservers.net --recv-keys 0x4E2C6E8793298290

2. (Optional).You can display information about this key by executing the following command:

gpg --fingerprint 0x4E2C6E8793298290

3.Verify that the private key corresponding to the public key you imported in Step 1 was used to generate the signature file that you downloaded in Step 0 (and that this signature file applies to the Tor Browser package) by executing the following command:

gpg --verify /path/public.key.tar.xz.asc /path/tor-browser-package.tar.xz

NOTA: As you can see, GPG displays a warning about the key used for this signature. This is because you have not actually verified the Tor Project's signing key. The best way to do this is to meet the Tor Project developers in person and ask them for the fingerprint of their signing key. For the purposes of this guide, we are relying on the fact that a well-known Tor Project GPG key (0x4E2C6E8793298290) was used to create a signature file that confirms the authenticity of the Tor Browser package that you downloaded.

The first time you launch the Tor Browser, it will ask you how it should connect to the Internet:

**Direct Access*: Select this option if your access to the Internet is unrestricted and if the use of Tor is not blocked, banned or monitored where you are located.

**Restricted Access*: Select this option if your access to the Internet is restricted or if the use of Tor is blocked, banned or monitored where you are located. If you want to use the Tor Browser from a location where the Tor network is blocked, you will have to use a bridge relay (https://bridges.torproject.org). Bridges are not listed in the public directory of Tor relays, so they are

more difficult to block. Some bridges also support pluggable transports, which try to disguise your traffic to and from the Tor network. This helps prevent online filters from identifying and blocking bridge relays. The default pluggable transport, called "obfs4", also makes it slightly more difficult for others to figure out that you are connecting to the Tor network. In general, though, Tor is not designed to hide the fact that you are using Tor.

NOTA: There are two ways to use bridges: you can enable the provided bridges or you can request custom bridges. If you are unable to access the Tor Project website, you can request custom bridge addresses by sending an email to *bridges@torproject.org* using a Riseup, Gmail or Yahoo account. Include the phrase, get bridges in the body of your message. If you can access the Tor Project website, you can obtain custom bridge addresses by visiting https://bridges.torproject.org/options and following the steps below. Read also https://www.torproject.org/docs/bridges , https://www.torproject.org/docs/pluggable-transports.html

After you configure the Tor Browser on the first launch, it will remember your selection and will not ask you to configure it again. You can change the configuration any time, from within the Tor Browser by clicking the "onion" button; this will activate the Tor Browser options menu. This might be necessary if you are travelling or if the situation changes in your country.

Once you already have Tor Browser running, you can verify you're using the Tor network going to https://check.torproject.org site. Moreover, you can go to any site informing about your public IP (for instance, https://www.iplocation.net or https://www.ip2location.com among many others) to see if it is faked

You can create a "new identity" for your Tor Browser anytime, too. When you do, the Tor Browser will randomly select a new set of Tor relays, which will make you appear to be coming from a different IP addresss again .To do this, click the "onion" button and select "New Identity" from the menu; the Tor Browser will clear your browsing history and cookies and then restart; once the it has restarted, you can confirm that you appear to be coming from a new IP address as described in the previous paragraph.

Software complementario

El proyecto Tor desarrolla otros programas complementarios del Tor Browser, los cuales se listan en https://www.torproject.org/projects/projects.html.en . Los más destacados, de todas formas, son:

https://guardianproject.info/apps/orbot : Tor on the Google Android mobile operating system. A related application is Orlib; a library for use by any Android application to route Internet traffic through Orbot/Tor.

https://nyx.torproject.org : Nyx is a terminal status monitor for Tor, intended for command-line aficionados and ssh connections. This functions much like top does for system usage, providing real time information on Tor's resource utilization and state

https://stem.torproject.org : Python library for applications and scripts that interact with Tor.

https://www.torproject.org/docs/pluggable-transports.html.en : Pluggable Transports (PT) transform the Tor traffic flow between the client and the bridge. This way, censors who monitor traffic between the client and the bridge will see innocent-looking transformed traffic instead of the actual Tor traffic.

https://ooni.torproject.org : Global observation network which aims is to collect high quality data using open methodologies, using free software to share observations and data about the various types, methods, and amounts of network tampering in the world

https://metrics.torproject.org : Analytics for the Tor network, including graphs of its available bandwidth and estimated userbase. This is a great resource for researchers interested in detailed statistics about Tor

https://metrics.torproject.org/onionoo.html : Web-based protocol to learn about currently running Tor relays and bridges

http://shadow.github.io : Network simulator that runs the real Tor software as a plug-in. Shadow is open-source software that enables accurate, efficient, controlled, and repeatable Tor experimentation. Another similar tool is Chutney (https://gitweb.torproject.org/chutney.git)

Configuración del acceso a la red Tor desde cualquier programa

Si queremos que más programas instalados en nuestro sistema puedan acceder a la red Tor, más allá del "Tor Browser", deberemos emplear el software Tor propiamente dicho (https://dist.torproject.org). Tras instalar este software (ya sea directamente a partir del paquete homónimo del repositorio de la distribución que usemos, del paquete disponible en el repositorio propio de Tor -ver cuadro más abajo- o bien a partir de la compilación del código fuente descargado del enlace anterior) podremos poner en marcha un servidor proxy local mediante los comandos habituales: *systemctl start tor, systemctl enable tor,* etc. A partir de entonces, cualquier programa que soporte proxys de tipo SOCKS podrá acceder a la red anónima.

NOTA:El protocolo SOCKS es una tecnología genérica de proxificación TCP de la cual el protocolo TOR es un caso particular.

NOTA:Puede que el programa que usemos incluya nuestra IP en los datos que envía, y entonces todo el enrutado y los cifrados que hagamos no servirán para nada. Por eso en general se suele recomendar el uso de "Tor Browser", ya que estos aspectos ya los tiene resueltos, en vez de configurar manualmente el proxy Tor.

To add official Tor repository in a Ubuntu system you must execute following commands (as root):

1.-*echo "deb https://deb.torproject.org/torproject.org bionic main" > /etc/apt/sources.list.d/tor.list*

2.-gpg –keyserver keys.gnupg.net –recv A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89
gpg --export A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89 | apt-key add -

3.-*apt update && apt install tor deb.torproject.org-keyring*

Concretamente, el programa en cuestión deberá configurarse para utilizar un proxy de tipo SOCKS. Tor solo escucha por defecto en la IP 127.0.0.1 y en el puerto 9050 (aunque si usamos Tor Browser este también puede actuar como proxy de otras aplicaciones, escuchando en este caso en el puerto 9150). Así pues, si queremos que un navegador Firefox "estándar" puede acceder a la red Tor, simplemente deberemos ir a "Preferencias→General→Servidor intermediario de red" y allí seleccionar "Ordenador central SOCKS (SOCKSv5)" indicando la IP de la máquina y puerto de escucha donde se está ejecutando el software Tor.

Algunes directives importants del fitxer de configuració del proxy Tor (/etc/tor/torrc) són:

**SOCKSPort {nº | ip:nº }* : Port (o combinació IP:port) per on escoltarà el proxy

**SOCKSPolicy {accept | reject } {ip | ipXarxa/mask | * }* : Regles que accepten o rebutjen connexions de clients amb una determinada IP o que provinguin d'una determinada xarxa o qualsevol. Es poden afegir les línies que es vulguin i la primera que quadri s'aplica i no es segueix llegint (a l'estil de l'Iptables). Per defecte s'accepten totes les connexions

**Log nivell /ruta/arxiu* : Indica el nivell de gravetat ("info", "warn", "error",…) a partir del qual es gravaran els missatges a l'arxiu de registre la ruta del qual s'hagi indicat

Si queremos usar una aplicación que no soporta proxificación SOCKS nativa, se pueden utilizar diferentes aplicaciones que añaden esta funcionalidad "on-the-fly" a cualquier aplicación, tal como *socat* (http://www.dest-unreach.org/socat), *proxychains* (https://github.com/haad/proxychains), *proxychains-ng* (https://github.com/rofl0r/proxychains-ng), *shadowsocks* (https://www.shadowsocks.org) o, más en concreto para la red Tor, *torsocks (*https://gitweb.torproject.org/torsocks.git) . Una llista encara més completa de

proxies es troba a https://en.wikipedia.org/wiki/Comparison_of_proxifiers Per més informació podeu consultar https://trac.torproject.org/projects/tor/wiki/doc/TorifyHOWTO

> NOTA: A vegades s'usa amb les eines anteriors un altre programa anomenat *Privoxy* (http://www.privoxy.org). Aquest és proxy HTTP, no pas TCP (és a dir, és més similar a Squid). Està especialitzat en augmentar la privacitat de la comunicació web gràcies a les capacitats de filtratge que incorpora (modificant les capçaleres HTTP, controlant l'accés, treient anuncis/ad-ware, no catxejant cap connexió, etc) però només serveix per comunicacions HTTP. Si s'enllaça Tor amb Privoxy podrem fem que llavors totes les connexions estiguin "torificades" i que les de tipus web, a més, passin per Privoxy.

La manera d'utilitzar el client Torsocks és simplement, des d'un terminal, escriure: *torsocks nomPrograma* Algunes directives importants del seu fitxer de configuració (/etc/tor/torsocks.conf) són *TorAddress ip (*IP del proxy Tor on es connectarà -per defecte 127.0.0.1-) i *TorPort nº* (Port del proxy Tor on es connectarà -per defecte 9050-), però en té moltes més; podeu consultar la seva documentació a la pàgina del manual o bé a https://trac.torproject.org/projects/tor/wiki/doc/torsocks

Implementación de servicios ocultos ("onion")

En la red Tor también pueden haber servidores web, SSH, mensajería, etc. No obstante, estos servidores solamente están disponibles dentro de la propia red Tor, no desde fuera. Por eso se les suele llamar servicios "ocultos". Como en la red Tor no existen ningún sistema centralizado de nombres tipo DNS, para conseguir que un servicio oculto esté disponible para los demás nodos lo que hace es crear varios "puntos de introducción" en ciertos nodos de la red, y notifica anónimamente a una base de datos replicada (los "directory servers") qué nodos son (en forma de lista, llamada "onion service descriptor"). Cuando un cliente quiera conectarse, enviará a uno de esos nodos la dirección de un determinado punto de encuentro (al que está conectado) y una clave única. El punto de introducción conectará con el servicio oculto, que se conectará al punto de encuentro, estableciendo así una comunicación entre el cliente y el servicio. La forma en la que están planteados estos servicios ocultos permite conectarnos a servidores de correo o de chat sin saber ni siquiera su dirección exacta, usando intermediarios y circuitos Tor anónimos. De hecho, es prácticamente imposible rastrear el envio de un correo: ni siquiera el propio servidor de correo sabe con qué ordenador estaba comunicándose. Para saber más sobre los detalles técnicos sobre cómo funcionan y se localizan los servicios "onion" recomiendo leer https://www.torproject.org/docs/onion-services.html.en

Para ofrecer un determinado servicio en la red Tor solo hay que editar el archivo /etc/tor/torrc. Concretamente, hay que añadir las siguientes líneas (y reiniciar el servicio)...:

HiddenServiceDir /var/lib/tor/hidden_service/
HiddenServicePort 80 127.0.0.1:8090

...donde en este caso concreto estamos redireccionando el puerto 80 de nuestra máquina (que es donde los clientes de la red Tor irán a conectarse a dicho servidor) a la IP loopback y puerto 8090 de la misma (que es donde se supone que está escuchando realmente nuestro servidor). La primera línea indica la carpeta (con permisos de lectura y escritura para el usuario que ejecute el proxy Tor) donde se almacenarán dos ficheros (creados al automáticamente al iniciar el proxy Tor): una clave privada (que identificará al servicio en cuestión) y un fichero llamado "hostname" cuyo valor contenido en su interior (consistente en un conjunto de caracteres ininteligibles -en realidad, un resumen de la clave pública asociada al servicio- finalizados por el sufijo ".onion") será el nombre "de dominio" que deberemos utilizar en el cliente adecuado (navegador, cliente SSH, mensajería, etc) para acceder, dentro de la red Tor, a ese servidor. La clave privada permite que dicho nombre sea siempre igual; de hecho, simplemente copiando el contenido de la carpeta /var/lib/tor/hidden_service a otra máquina (y teniendo el fichero torrc con la misma configuración también) se puede continuar ofreciendo el mismo servicio con el mismo nombre como si no hubiera pasado nada.

Para cada servicio que queramos poner en marcha en la red Tor deberemos de indicar las dos líneas anteriores en ese orden: primero *HiddenServiceDir* y luego *HiddenServicePort*

> NOTA: If you set *HiddenServiceAuthorizeClient* too, then it is only available for authorized clients.

Implementación de "relays"

Por defecto Tor actúa solamente como cliente de entrada a la red. Si se quiere ayudar a fortalecer y agrandar la red Tor, se puede convertir nuestro nodo Tor en un nodo interno (un "relay") para permitir que tráfico de otros clientes atraviesen nuestro nodo como un punto más de su itinerario. Esto se logra teniendo en el archivo /etc/tor/torrc de un sistema operativo con IP pública (que puede ser uno contratado, por ejemplo, en Linode, DigitalOcean o AWS) la siguiente línea:

* *ORPort {nº|auto}*: Informa a los clientes de la red Tor de su presencia para que lo utilicen. Se puede indicar el número de un puerto concreto para las peticiones de los clientes (normalmente se elige el 9001) o bien la palabra "auto" para que se elija un puerto al azar automáticamente.

Como medida suplementaria de seguridad, se suele establecer además la línea *SocksPort* a 0 para evitar que nuestro nodo sea un nodo de entrada. Otras líneas, opcionales, son además *Nickname nom* o *Contact-Info direccio@corr.eu*

También se puede indicar que deseamos que nuestro nodo Tor se convierta, además de un nodo interno, en un nodo de salida (un "output relay") para así permitir que tráfico de otros clientes utilicen nuestro nodo también para salir a Internet. Hay que tener cuidado con esto porque en este caso las actividades que realicen dichos clientes aparecerán como que las está realizando nuestro nodo de salida. Esto se consigue añadiendo al archivo /etc/tor/torrc, además de la directiva ORPort, la línea *ExitRelay {0|1}* (donde 1 activaría la funcionalidad de "exit relay" y 0 -valor por defecto-, no.

Igualmente, también podríamos configurar nuestro nodo Tor como un "directory server" mediante la directiva *DirPort {nº | auto }* Para más información, https://www.torproject.org/docs/tor-manual.html.en i, en general, a https://trac.torproject.org/projects/tor/wiki/TorRelayGuide i https://blog.torproject.org/tips-running-exit-node

Hay que tener en cuenta, en cualquier caso, que para que nuestro nodo sea reconocido efectivamente como un "relay" válido (ya sea interno, de salida o de directorio) primero las Autoridades deberán alcanzar el Consenso para ello.

Otras redes

Tor no es la única red "oculta" que existe en Internet. A continuación listamos otras redes ocultas que son importantes. Los protocolos internos de cada una son diferentes pero todas ellas funcionan en apariencia de forma similar: debemos ejecutar en nuestro ordenador un determinado software que nos da acceso a la red oculta, dentro de la cual podemos disponer de varios servicios, entre los cuales es "salir a la superfície" de Internet de nuevo de forma anónima.

I2P (https://geti2p.net/en)
FreeNet (https://freenetproject.org)
GNUNet (https://gnunet.org)
ZeroNet (https://zeronet.io)
LibreNet (http://librevpn.org.ar)
ChaosVPN (https://wiki.hamburg.ccc.de/ChaosVPN)
Lantern (https://getlantern.org) -Aplicació específica para esquivar la censura-

Por otro lado también conviene destacar IPFS (https://ipfs.io), el cual es una red de nodos que no actúan como elementos independientes sino como sistema P2P global de ficheros en red. Esto permite alojar contenido en esta red de una forma distribuida y, por tanto, muy difícil de censurar y eliminar.