Universidad Técnica Federico Santa María
Departamento de Electrónica
ELO322 - Redes de Computadores I

# Proyecto Redes de Computadores

## Redes de anonimización Tor

19 de septiembre de 2015

Margarita Guerrero Salazar
Mario López Cáceres

# Índice



# Índice de figuras

## Abstract

Las redes de anonimización Tor es un sistema bien útil para ocultar la ubicación y los mensajes emanados desde un cliente. Con varios mecanismos para combatir la censura, se plantea como una solución real a tal problema. En el presente trabajo se busca abordar las características de los circuitos virtuales Tor y cómo se crean. Se comprueba finalmente la anonimidad ante la Internet Superficial y la posibilidad de saltar bloqueos regionales.

# 1. Introducción

Una de las características estudiadas respecto a la comunicación vía Internet, es la exposición pública de la IP origen y destino de los paquetes involucrados en el proceso. En este punto, solamente está al alcance de los comunicantes la encriptación y confidencialidad de los datos intercambiados; mas no sus direcciones IP. Lo anterior abre la posibilidad, para un intruso presente en cualquier punto del camino de datos, localizar las ubicaciones físicas de los comunicantes terminales con suma facilidad. Ante esta situación, y por diversos que sean los motivos o intenciones de camuflar tanto el mensaje, como los emisores y receptores; surge la red de anonimato *The Onion Router* (Tor).

Para tener un breve contexto histórico, la filosofía que comprende el funcionamiento de Tor, el *enrutamiento cebolla*, surgió a mediados de los años 90, por el matemático Paul Syverson, el informático teórico Michael Reed y David Goldschlag, empleados del Laboratorio de Investigación Naval de los Estados Unidos. El propósito de la investigación era salvaguardar las comunicaciones de inteligencia estadounidenses en Internet. Posteriormente el concepto fue desarrollado por DARPA. El 20 de septiembre de 2002 fue lanzada una versión alpha del programa, llamando al proyecto *The Onion Router project*, o proyecto TOR. En 2004 el Laboratorio de Investigación Naval de Estados Unidos libera el código de Tor bajo licencia gratuita. En el mismo año, la Electronic Frontier Foundation financia a los creadores de la versión alpha, Paul Syverson y Roger Dingledine, para continuar su desarrollo. Actualmente, el desarrollo y revisión de Tor, se encuentra en manos de la organización sin fines de lucro *The Tor Project, Inc.*, radicada en Massachusetts [1].

En el presente informe se plantea como objetivo comprender en términos generales en qué consiste el enrutamiento cebolla, qué elementos lo conforman, cómo se estructura y organiza la red Tor; finalmente se busca efectuar un experimento práctico para navegar en las redes Tor y observar cómo los sitios de la *Internet superficial* interactúan con la misma.

# 2. Conceptos previos

## 2.1. Encriptación

Antes de estudiar el enrutamiento de cebolla, es necesario tener una noción sobre encriptación. Ello permitirá entender cómo se establecen las rutas en la red Tor, la confidencialidad y el cifrado de datos. Primero se tratará la criptografía de clave pública.

### 2.1.1. Criptografía de clave pública

Se considerarán a dos usuarios: Patricio y Lisa. Lisa quiere enviar un mensaje a Patricio, que únicamente él será capaz de leer (descifrar). Patricio dispone de dos *claves*, una *clave pública* y una *clave privada*. La clave pública es conocida por Lisa y cualquier otro usuario de la red, a excepción de la clave privada, que es solamente conocida por Patricio. Así, la comunicación procede de acuerdo a los siguientes pasos:

- Lisa escribe un mensaje.
- Lisa, utilizando la clave pública, "transforma"(algoritmo de encriptación) el mensaje. Los datos resultantes son irreconocibles del mensaje original, ni si quiera Lisa (y cualquier otro) es capaz de reconocer el mensaje original.
- Se envía el mensaje encriptado por Internet a Patricio.
- Patricio, utilizando la clave privada, "transforma"(algoritmo de desencriptación) el mensaje, siendo los datos resultantes el mensaje original que escribió Lisa logrando confidencialidad.

Es requisito a los algoritmos mencionados, que para cualquier mensaje, sólo sea válida una única clave privada, para una única clave pública (análogo a la biyección en funciones). Para ilustrar con un método real, sin entrar en detalles de su demostración para no desviar el tema central del presente informe, se enuncia el algoritmo del sistema RSA [3]:

"Primero se eligen dos números primos $p$ y $q$ y se calcula $N = pq$ y $\phi = (p-1)(q-1)$. Se elige aleatoriamente un número entero $e$ entre 3 y $N-2$ tal que $e$ y $\phi$ no tienen factores primos comunes. La clave pública es el par $(N, e)$. La clave privada $d$, que el emisor guardará en secreto, se calcula de tal manera que $d \equiv e^{-1}$ (mód $\phi$), esto es, el número tal que $de \equiv 1$ (mód $\phi$)."

La efectividad del método anterior recae en el hecho de que a partir de un número $N$, es computacionalmente inviable por medio de fuerza bruta calcular los factores primos $p$ y $q$. Esto es debido al uso de números de 309 (RSA-1024 bits) y 617 dígitos (RSA-2048 bits), estimando un tiempo de éxito o ruptura de órdenes astronómicos (varias edades del universo). Una de las falencias de este método es la certeza o seguridad de que la clave pública que se conoce, sea realmente la correcta (autenticación de la fuente que provee la clave pública).

#### 2.1.2. Protocolo de establecimiento de claves

A diferencia del punto anterior, el presente contexto consiste en dos usuarios que desean establecer una clave común, sin que un tercero con acceso a la conversación, pueda deducirla. En el caso particular de Tor, el protocolo de interés corresponde al denominado Diffie-Hellman.

Supóngase los mismos protagonistas del item anterior: Patricio y Lisa. No se ha establecido contacto previo y el canal se encuentra comprometido (inseguro):

- Patricio y Lisa concuerdan abiertamente (públicamente) en utilizar el número primo $p$ y la base $g$.
- Patricio elige un número aleatorio $a < p$, al igual que Lisa elige su propio número aleatorio $b < p$.
- Patricio calcula, utilizando su número secreto, $A = g^a \ mod \ p$ y se lo envía públicamente a Lisa. Lisa calcula, también utilizando su número secreto, $B = g^b \ mod \ p$ y se lo envía públicamente a Patricio.
- Patricio calcula usando su número secreto $K = B^a \ mod \ p$. Lisa calcula usando su número secreto $K = A^b \ mod \ p$.

Finalmente ambos calculan y obtienen el mismo número $K$, sin haber compartido directamente tal. Para un intruso, la única información que dispone son los números $p$, $g$, $A$ y $B$. No obstante, necesita ya sea el número secreto $a$ o $b$. En teoría, si desea obtener $a$ (o $b$), debe resolver qué número, al usarse como exponente de $g$ y calculando el módulo $p$, se obtiene $A$. La resolución anterior es inviable computacionalmente por medio de fuerza bruta.

Una importante ventaja de éste método es que, en el caso teórico de que se rompa por fuerza bruta la clave secreta, ambos interlocutores tienen la posibilidad de efectuar una nueva negociación que deje inválida la clave anterior. De ésta forma se pueden hacer negociaciones periódicas que actualicen la clave secreta, limitando bastante el tiempo disponible para un ataque de fuerza bruta.

Se debe considerar que una falencia de este método es la posibilidad de un ataque *Man-in-the-Middle*, donde un atacante ubicado entre ambos interlocutores intercepta los mensajes y negocia con cada uno un proceso independiente de Diffie-Hellman, sin que ambos se enteren. Por lo anterior, es recomendable utilizar un método mixto entre Diffie-Hellman y de verificación de identidad.

# 3. Red Tor

Teniendo un acercamiento al concepto de encriptación, se está en condiciones de estudiar cómo se establece y trabaja la red Tor, empezando por la idea fundamental: Enrutamiento de cebolla.

## 3.1. Enrutamiento de cebolla

Cuando se conforma un circuito virtual en la red Tor, participan de ello (a lo menos) 3 nodos intermedios:

- Nodo de entrada: Éste es el punto de entrada a la Red Tor.
- Nodo intermedio: Establece un punto de separación entre el nodo de entrada y el nodo de salida, para prevenir que ambos nodos terminales tengan conocimiento el uno del otro.
- Nodo de salida: Son los puntos de salida de la red Tor. Intermedian entre la red Tor y la *Internet superficial*, envían el mensaje final al punto de destino.

Cuando un host desea establecer un circuito virtual en la red Tor, accede a un servidor donde hay una lista de las direcciones de nodos (no todos) y sus llaves de encriptación públicas. Con las llaves públicas, el host puede encriptar un mensaje y ser desencriptado únicamente por el nodo dueño de la llave (usado únicamente para la negociación) [2].

Para crear el circuito virtual, se procede similar al envío de un mensaje:

- El host, encriptando los mensajes enviados por medio de la clave pública, negocia y genera una clave privada entre él y el nodo de entrada por medio de Diffie-Hellman (Clave A).
- Usando como túnel al nodo de entrada (encriptado bajo la clave A), el host negocia una nueva clave privada (Clave B) por medio de Diffie-Hellman con el nodo intermedio. El mensaje de negociación se encuentra encriptado con la clave pública del nodo intermedio.
- Usando nuevamente como túnel la conexión **host-nodo de entrada-nodo intermedio** (encriptado con la Clave A y B), se negocia una última clave con el nodo de salida (Clave C).

De esta forma, cada nodo comparte una clave de encriptación privada con el host. Para enviar un mensaje, éste se encripta primero con la Clave C (nodo de salida). Éstos datos son nuevamente encriptados con la Clave B (nodo intermedio), y finalmente son encriptados con la Clave A (nodo de entrada). La estructura del mensaje se puede ilustrar como las capas de una cebolla:

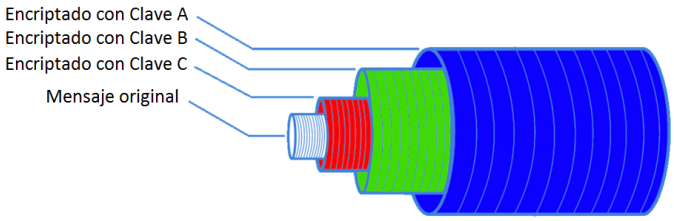


Fig. 1: Estructura de un mensaje en la red Tor.

Así, cada nodo desencripta (desenvuelve) los datos enviados, limitándose a identificar la información corespondiente a la fuente y destino próximos (nodo anterior y siguiente), de esta forma un nodo individual no puede conocer la ruta completa del circuito virtual. No obstante, el nodo de salida tiene libre acceso al contenido del mensaje al ser el nodo que desenvuelve la última capa. Por ello utilizar protocolos sin encriptación (HTTP, texto plano, etc.) otorga la posibilidad de que el nodo de salida pueda hurgar en los datos enviados [5].

Cuando se retorna un mensaje desde la Internet superficial al nodo de salida, se efectúa el proceso inverso: Cada nodo encripta el mensaje recibido del nodo anterior con la clave compartida con el host. Finalmente el host desencripta capa por capa los datos recibidos hasta obtener el mensaje de respuesta.

## 3.2. Servicio de directorio: El Consenso

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 *El Consenso* [7].

En cada cliente Tor se encuentra embebido la información de 10 nodos **confiables**. La misión de estos nodos es de actualizar y mantener El Consenso, por lo cual son llamados *Autoridades de directorio* (Directory authorities). Sus nombres y direcciones son:

- Autoridades de nodos: item moria1/128.31.0.39, tor26/86.59.21.38, Faravahar/154.35.32.5, urras/208.83.223.34, dannenberg/193.23.244.244, maatuska/171.25.193.9, longclaw/199.254.238.52, dizum/194.109.206.212 y gabelmoo/131.188.40.189.
- Autoridad de bridges: Tonga/82.94.251.203.

Las autoridades de nodos se encargan de actualizar El Consenso. 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 autoridades.
- 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. Por ejemplo, la versión mas actualizada del consenso (versión de tor26) se puede obtener como texto plano desde la dirección http://86.59.21.38/tor/status-vote/current/consensus/. No obstante, el proceso de actualización del consenso es efectuado por 9 autoridades; la décima autoridad, Tonga, es ajena a ello. La función de Tonga es la de mantener una lista con información sobre los *nodos bridges* (puente).

## 3.3. Nodos bridges o repetidores puente

Para entender uno de los problemas que debe sortear el proyecto Tor, se considera la existencia de un gobierno opresivo. Éste gobierno, dado que tiene acceso a la lista pública de nodos de la red Tor y con ello sus direcciones IP, es capaz de bloquear el acceso de sus ciudadanos a cada nodo. Así, surge la solución a este problema por medio de los *nodos bridges* bajo el proyecto *BridgeDB* [6].

BridgeDB otorga la información (provista por Tonga) de unos pocos repetidores puente a la vez y cada cierto tiempo. Se debe acceder a https://bridges.torproject.org/bridges, y resolviendo un *Captcha* se obtiene la información de 3 bridges aleatorios. Además, existen modos alternativos para solicitar direcciones de repetidores puente, como enviar un correo electrónico a bridges@torproject.org o pedir ayuda a help@rt.torproject.org en caso de mal funcionamiento.

De esta forma, usando estos nodos bridges como nodos de entrada, se puede saltar el bloqueo y navegar de forma normal en la red Tor, ya que las conexiones de los nodos puentes al resto de nodos se ven fuera del alcance del gobierno opresor.

También se obtiene la ventaja de que, dado que no todos los nodos son públicos, un nodo de entrada al negociar con un cliente, no es capaz de discernir si efectivamente el cliente es un host o es un nodo puente. De esta forma, el nodo de entrada no puede conocer su ubicación en el circuito virtual (si es nodo de entrada o intermedio).

## 3.4. Software disponible

Para facilitar el acceso a las redes Tor, o participar del proyecto donando ancho de banda como nodo de la red, el proyecto Tor pone a disposición del usuario el software *Tor Browser*. El programa contiene una versión modificada del navegador Firefox, TorLauncher, TorButton, NoScript, entre otros. Al momento de ejecutar el navegador, inmediatamente corre en segundo plano los procesos de enrutamiento y creación de los circuitos virtuales.

También se encuentra disponible el programa *Vidalia*. Vidalia constituye una interfaz gráfica para monitorizar los circuitos virtuales creados por el navegador Tor, los nodos de la red y configurar otros aspectos relacionados con la participación como nodo en la red. Es importante señalar que no es parte del proyecto Tor, siendo una aplicación creada por un tercero, además de encontrarse actualmente descontinuada (Incompatible para versiones 3.x en adelante del navegador Tor, reemplazada parcialmente por la extensión TorLauncher).

Finalmente, cabe mencionar la existencia del sistema operativo *Tails* (The Amnesic Incognito Live System). Éste se encuentra basado en Debian GNU/Linux, el cual viene con el navegador Tor y Vidalia preinstalados; diseñado para no rastros de uso en el PC utilizado, dado que corre desde un LiveCD o pendrive.

# 4. Acceso a la red Tor

Utilizando el programa VirtualBox, se creó una máquina virtual en la cual se corrió el sistema operativo Tails, y con ello se experimentó con el acceso a las redes Tor.

## 4.1. Reconocimiento del enrutamiento de cebolla

Para validar el funcionamiento del enrutamiento de cebolla, se accedió desde Tor al sitio *https://www.whatismyip.com* para observar cuál era la IP que reconocía el sitio como IP origen:

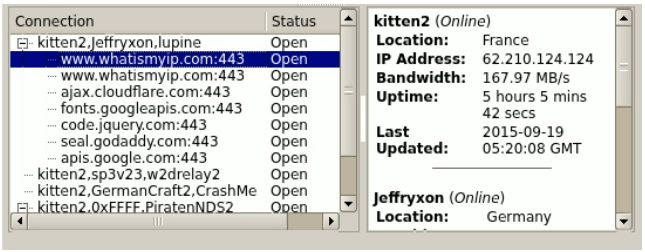


Fig. 2: Interfaz Vidalia: Información del nodo de entrada

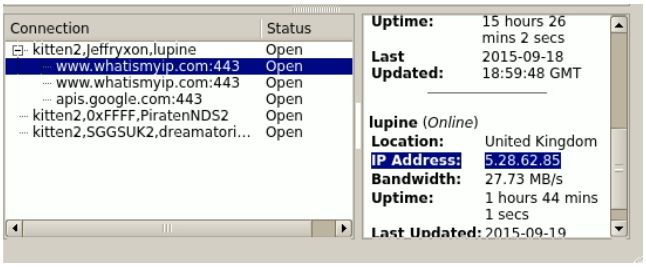


Fig. 3: Interfaz Vidalia: Información del nodo de salida

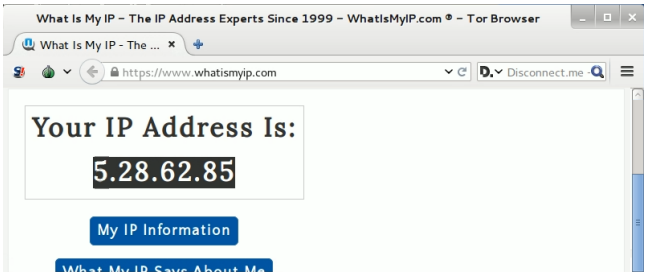


Fig. 4: IP de origen reconocida por el servidor de https://www.whatismyip.com.

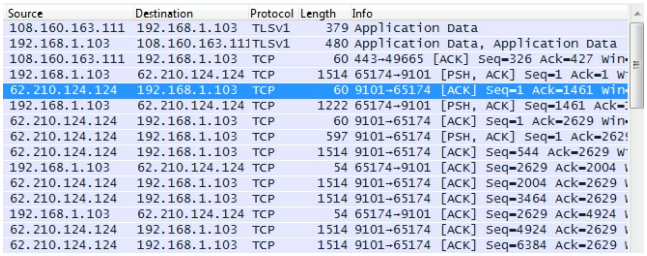


Fig. 5: Captura de Wireshark: Conexión directa con kitten2.

Por medio de la interfaz de Vidalia, se identificó en el mapa de red el circuito virtual usado, implementado en los nodos **kitten2-Jeffryxon-lupine** como entrada (IP: 62.210.124.124), intermedio y salida (IP: 5.28.62.85) respectivamente.

Al observar lo indicado por el sitio web, éste reconoció la IP de origen 5.28.62.85, la cual corresponde efectivamente a la del nodo de salida lupine. En el caso del nodo de entrada, se comprobó con Wireshark que para acceder al sitio, la conexión y desde donde se descargaron los datos fue desde la IP 62.210.124.124 (nodo kitten2). En los paquetes no se encontraron segmentos TCP relacionados con el establecimiento de la conexión, ya que la negociación se establece al formar el circuito virtual Tor. También es de notar que Wireshark no reconoció el requerimiento GET dado que los datos que llegan al computador se encuentran encriptados bajo las capas de cada nodo del circuito, siendo desencriptados por el navegador una vez que fueron descargados.

## 4.2. Acceso a contenido con bloqueo regional

Una ventaja interesante del enrutamiento de cebolla, es que dado que la internet superficial reconoce el acceso a su contenido desde el nodo de salida, se pueden acceder a contenidos bloqueados en la región o país del host por medio

de los circuitos virtuales Tor.

Se consideró el link del sitio YouTube *https://www.youtube.com/watch?v=dTaD9cd8hvw*, el cual se encuentra bloqueado en Chile por copyright. Por medio de Tor se establece un circuito virtual, con nodo de salida ubicado en Canadá; así, el sitio reconoce la conexión desde Canadá permitiendo la visualización del contenido:

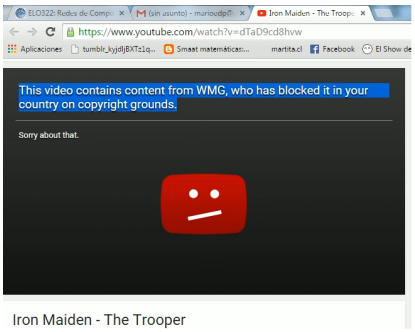


Fig. 6: Video bloqueado al acceder desde un navegador normal.

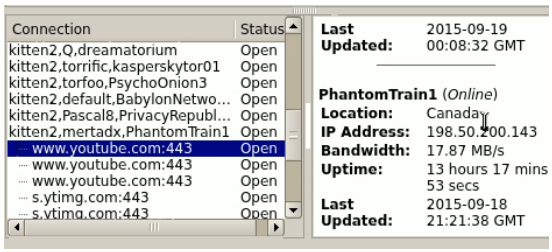


Fig. 7: Establecimiento de un circuito virtual con nodo de salida desde Canadá.


Fig. 8: Visualización exitosa del video al ser reconocido el acceso desde Canadá (país no bloqueado).

## 5. Conclusiones

Finalizando el presente informe, se abordaron conceptos relacionados a la encriptación que permiten entender y notar la posibilidad de la implementación del enrutamiento de cebolla. Se mostraron las características generales que mantienen a la Red Tor y cómo es que se crean los circuitos virtuales en tal red. Se observa además que la motivación del proyecto es la de confrontar la opresión y censura gubernamental, identificando para ello medidas adoptadas como la de nodos bridges; no obstante otros usos o posibilidades surgen de la misma red Tor.

Utilizando el software disponible por el proyecto Tor y por terceros, fue posible evidenciar y probar el funcionamiento del enrutamiento de cebolla y de cómo un cliente en la red Tor es visto por la Internet Superficial: bajo la identidad del nodo de salida. Esto da pie a varias posibilidades como se mencióno, tal como el acceso a contenido con bloqueo regional, el cual fue demostrado experimentalmente por medio del acceso a contenido protegido de YouTube, y otros.

Por límites de extensión, no fue posible abordar otro servicio importante de la red Tor, en el cual se sustenta el concepto de *Deep Web* (Internet Profunda), siendo aprovechado bastante para el mercado de sustancia ilícitas, intercambio de contenido prohibido y servicios ilegales: *Hidden Services* (Servicios Ocultos).

En conclusión y síntesis, considerando la extensión abordada en el presente informe de la red Tor, fue posible evidenciar su real funcionamiento y utilidad en la interacción con la Internet Superficial para usos domésticos, como es el acceso a contenido protegido y de cómo es posible ofrecer servicios de anonimización, sobre una red como Internet que no garantiza tal sobre sus comunicaciones.

# Referencias

[1] Wikipedia - **Tor (red de anonimato)**
https://es.wikipedia.org/wiki/Tor_(red_de_anonimato)

[2] Wikipedia - **Onion routing**
https://en.wikipedia.org/wiki/Onion_routing

[3] A. Nectoux - **Criptografía de clave pública**
http://blog.kleinproject.org/?p=1618&lang=es

[4] F. Campos - **El algoritmo de Diffie-Hellman**
http://www.javiercampos.es/blog/2011/07/22/el-algoritmo-de-diffie-hellman/

[5] J. Wright - **How Tor Works: Part One**
http://jordan-wright.com/blog/2015/02/28/how-tor-works-part-one/

[6] J. Wright - **How Tor Works: Part Two - Relays vs. Bridges**
http://jordan-wright.com/blog/2015/05/09/how-tor-works-part-two-relays-vs-bridges/

[7] J. Wright - **How Tor Works Part Three - the Consensus**
http://jordan-wright.com/blog/2015/05/14/how-tor-works-part-three-the-consensus/