Universidad de Buenos Aires
Facultades de Ciencias Económicas, Ciencias Exactas y Naturales e Ingeniería

Maestría en Seguridad Informática

Comunicaciones Secretas en Internet

Autor: Ing. Rafael Bonifaz
Director: Dr. Pedro Hecht

Año 2019
Cohorte 2016

Declaración Jurada de origen de los contenidos

Por medio de la presente, el autor manifiesta conocer y aceptar el Reglamento de Trabajos Finales vigente y se hace responsable que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual.

Rafael Bonifaz

Licencia

Resumen

El objetivo de este trabajo es mostrar la forma en la que dos o más personas se pueden comunicar de manera secreta a través de Internet sin que alguien más sepa que se comunicaron. En particular se investigó los casos de correo electrónico, chat y voz sobre IP.

La estructura del trabajo se divide en dos partes: en la primera, mediante la metodología de exploración bibliográfica, se analiza la capacidad de vigilancia en la red por parte de potencias como Estados Unidos y gobiernos latinoamericanos según filtraciones como las de Edward Snowden, Wikileaks y otras. Más adelante se analiza el uso de software libre, criptografía y autonomía como estrategias para ocultar las comunicaciones.

En la segunda parte se realiza una investigación sobre soluciones libres existentes mediante un análisis teórico y práctico. En primer lugar se analizan herramientas como VPNs, Tor e I2P para ver la forma en la que estas ayudan a ocultar la identidad de quiénes se comunican. En los capítulos subsiguientes se verá cómo el uso de Tor combinado con el cifrado entre extremos permitirá tener comunicaciones secretas para correo electrónico, chat y voz sobre IP

Palabras Claves: software libre, anonimato, privacidad, vigilancia, comunicaciones secretas, Internet

Contenido

Introducción....1
1 Vigilancia en Internet....3
1.1 Vigilancia Global....3
1.2 Vigilancia Estatal en América Latina....6
1.3 Modelo de Amenaza....10
2 Privacidad, Software Libre, Criptografía y Autonomía....12
2.1 Privacidad....13
2.2 Software Libre....15
2.3 Criptografía....17
2.4 Autonomía en las Comunicaciones....21
3 Redes de Anonimato y VPNs....23
3.1 Redes Virtuales Privadas - VPNs....24
3.2 La Red Tor....25
3.4 Análisis Anonimato....36
4 Correo Electrónico Secreto....38
4.1 Cifrado Extremo a Extremo con OpenPGP....39
4.2 Correo Secreto con Seudónimos....42
4.3 Anonimato y Cifrado con Cliente Externo....46
4.4 Correo Electrónico y Servicios Ocultos....47
4.5 Análisis Correo Electrónico....50
5 Chat Secreto....52
5.1 Anonimato con XMPP....54
5.2 Cifrado Extremo a Extremo con OTR....59
5.3 Signal y Derivados....61
5.4 Ricochet....66
5.5 Briar....68
5.6 Delta Chat....70
5.7 Análisis Chat....71
6 Llamadas de Voz sobre IP Secretas....73
6.1 Cifrado Extremo a Extremo y Voz sobre IP....74
6.2 Tox y Ring....76
6.3 Comunicaciones Secretas con Mumble....78
6.4 Análisis de Llamadas Secretas....81
7 Aportes Originales del Autor....83
Conclusiones....87
Anexo 1: Proveedores de Correo Electrónico....92
Anexo 2: Herramientas para Cifrado de Correo....93
Anexo 3: Clientes XMPP....94
Bibliografía....95
Índice de Imágenes....101

Introducción

Las comunicaciones en Internet son vigiladas por agencias de inteligencia, gobiernos y corporaciones. Toda actividad en línea deja rastro y este puede ser asociado a la identidad de las personas a través de números telefónicos, direcciones IP, cuentas de correo electrónico, cuentas de redes sociales, localización u otros metadatos. El presente trabajo tiene como objetivo mostrar como dos o más personas se pueden comunicar entre sí de manera secreta sin que nadie más sepa que esa comunicación existió ni el contenido de la misma. Hay muchas formas de comunicarse en Internet por lo que se realiza el análisis sobre las comunicaciones de correo electrónico, chat y voz sobre IP.

En el primer capítulo se verá los alcances de la vigilancia a nivel global a través de los programas revelados por Edward Snowden, Wikileaks y otros. Se presentan casos de espionaje político realizados por gobiernos de América Latina a través de las tecnologías de comunicación. Esto servirá para identificar el modelo de amenaza sobre el que se quiere proteger la comunicación.

En el segundo capítulo se explica que es la privacidad y por qué es importante defender este derecho humano. A continuación se hace un análisis sobre la importancia del software libre, la criptografía y la autonomía para llegar al objetivo de las comunicaciones secretas.

En el tercer capítulo se muestran las herramientas disponibles para tener anonimato en Internet. Del cuarto al sexto capítulo se analizan herramientas para tener comunicaciones cifradas extremo a extremo para correo, chat y voz sobre IP. En los tres casos se combinan estas herramientas con la red de anonimato Tor para ocultar los metadatos de las comunicaciones.

El capítulo siete detalla los aportes del autor en este trabajo. Los mismos tienen que ver con fomentar una visión crítica al uso de la tecnología con una visión política. En este documento se muestra como las

comunicaciones son vigiladas a nivel global y estatal; a pesar de esto el autor muestra estrategias viables para tener comunicaciones secretas. Para que las comunicaciones sean secretas, no basta con ocultar el contenido de las mismas, hay que ocultar el hecho de que estas existieron. Por último los aportes van más allá de lo teórico y gran parte de las pruebas de este trabajo se realizaron en colaboración con personas interesadas o mediante capacitaciones.

En la conclusión se hace un análisis sobre las pruebas realizadas. Si bien se pudo establecer comunicaciones secretas, en la mayoría de los casos esto no fue un trabajo sencillo y se requieren conocimientos técnicos. Adicionalmente se hacen recomendaciones de lo que se debería hacer para mejorar la privacidad en las comunicaciones.

1 Vigilancia en Internet

Las comunicaciones en Internet pueden ser vigiladas por agencias de inteligencia de potencias como los Estados Unidos o por estados más pequeños como es el caso de Latinoamérica. En este capítulo se expondrán de manera breve varios casos de espionaje estatal de los que se tiene conocimiento.

En la sección 1.1 se explica los alcances de vigilancia masiva realizado por los Estados Unidos y sus aliados según las revelaciones de Edward Snowden. En la sección 1.2 se muestran algunos casos de espionaje realizados por gobiernos de América Latina. Se finaliza el capítulo con la sección 1.3 donde se explica el modelo de amenaza sobre el cual se trabajará a lo largo del trabajo.

1.1 Vigilancia Global

Para el “Trabajo Final de Especialización en Seguridad Informática”, el autor realizó una investigación sobre las capacidades de vigilancia de La Agencia Nacional de Seguridad de los Estados Unidos (NSA por sus siglas en inglés).[1] En junio de 2013 Edward Snowden, ex contratista de la NSA, filtró miles de documentos a los periodistas Glenn Greenwald y Laura Poitras. Los mismos contienen información interna que muestra las capacidades de la NSA para recolectar comunicaciones de Internet de forma masiva y cómo procesa toda la información almacenada.

Una de las revelaciones más importantes de Snowden es el programa PRISM. En una presentación de 2013 se explica que, para entonces, en este programa participaban las empresas Microsoft, Google, Yahoo, Facebook, Paltalk, Youtube, Skype, AOL y Apple; también se anunció que Dropbox se sumaría como proveedor de información para PRISM.

Según los documentos, las corporaciones que participan en este programa entregan los datos de sus usuarios al gobierno de Estados Unidos cuando este se lo solicita. Salvo que una persona sea ciudadana de Estados Unidos y se encuentre en este país, no existe ningún tipo de protección legal. Es decir que agencias como la NSA, CIA y el FBI pueden acceder a los datos de ciudadanos extranjeros almacenados en los servidores de estas empresas sin ser informados y sin necesidad de una orden judicial.

Entre la información que se entrega a estas agencias se encuentra correos electrónicos, conversaciones de chat, videos, llamadas de voz, entre otros como se puede ver en la imagen 1. En otra diapositiva de esta misma presentación se afirma que PRISM es la principal fuente de la NSA para generar informes de inteligencia. [2]

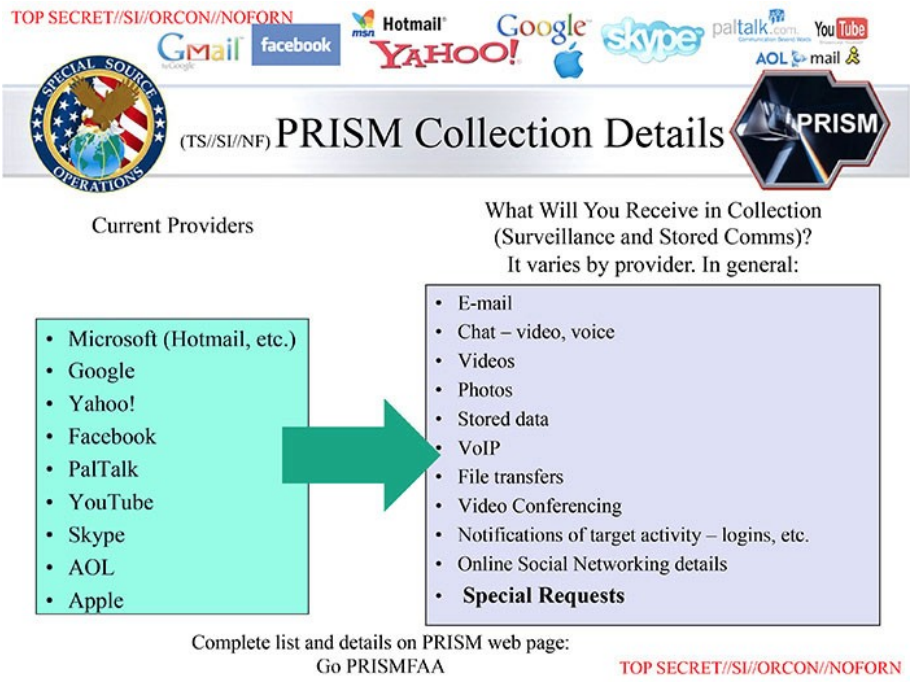

Imagen 1: Diapositiva de presentación sobre PRISM de 2013
Fuente: https://edwardsnowden.com

PRISM es una de las muchas formas que tiene la NSA para recolectar información de Internet. En la imagen 2 se puede ver una diapositiva que muestra que en 2012 la NSA recolectaba información con el apoyo de agencias de inteligencia de otros países (3rdPARTY/LIASON), interceptaba comunicaciones celulares desde misiones diplomáticas (REGIONAL), a través de ataques informáticos (CNE), directamente desde

los cables de fibra óptica (LARGE CABLE) e interceptando las comunicaciones satelitales (FORNSAT).[3]

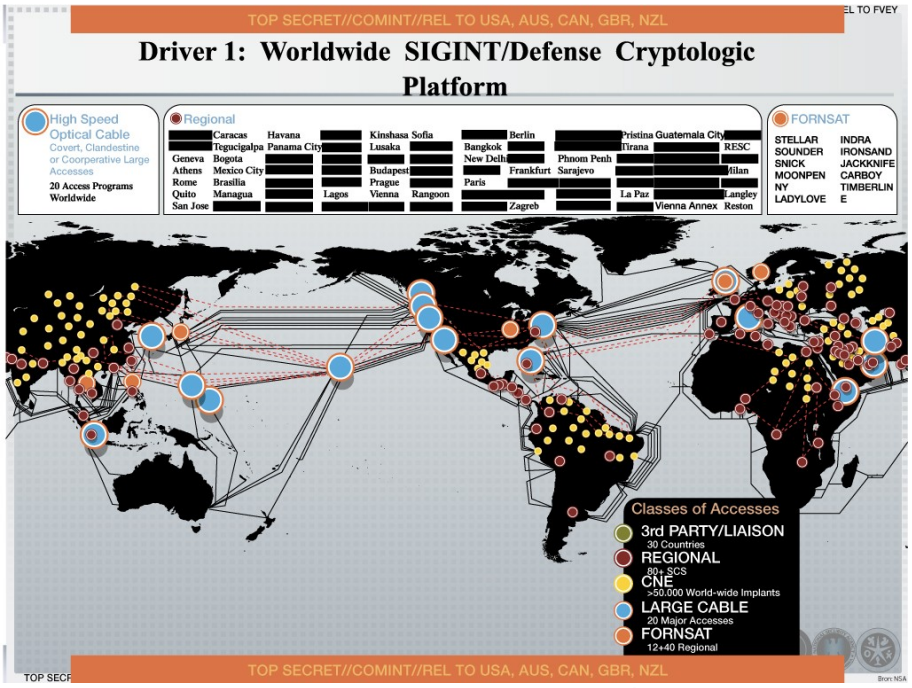

Imagen 2: Formas en que la NSA recolecta información
Fuente: https://edwardsnowden.com

La información recolectada luego es procesada en varios sistemas que permiten realizar búsquedas. El sistema XKEYSCORE funciona de manera similar a un buscador de Internet sobre los datos recolectados por la NSA. Para 2009, en este sistema se podía realizar búsquedas de hasta tres días de contenido de las comunicaciones y treinta días de metadatos[1] de todo el tráfico recolectado de forma masiva por la NSA. Otros sistemas como Pinwale permiten almacenar datos por cinco años.[1]

Snowden describe las capacidades de estos sistemas: "Yo, sentado en mi escritorio, tenía la facultad de intervenir al que fuera, desde un contador hasta un juez federal e incluso el presidente, siempre y cuando tuviera su correo electrónico personal." [4]

La NSA no es la única agencia de Estados Unidos que espía en Internet. Revelaciones realizadas por Wikileaks en el año 2017 muestran documentos de la CIA que ponen en evidencia la capacidad de esta agencia

1 Metadatos son los datos que describen a los datos.

para explotar fallos de seguridad en sistemas operativos de computadoras, celulares, televisiones inteligentes, entre otros.[5]

En diciembre de 2011 Wikileaks realizó la primera publicación de los documentos conocidos como Spyfiles. En esta entrega se denunció la existencia de 160 empresas, ubicadas en países tecnológicamente desarrollados, que venden productos de vigilancia a gobiernos a nivel mundial. Software como el de la empresa sudafricana Vastech permite grabar las llamadas telefónicas de naciones enteras, otras soluciones permite registrar la ubicación de todos los teléfonos celulares en una ciudad. Empresas como la italiana Hacking Team o la francesa Vupen venden software para instalar *malware* en teléfonos celulares y tomar control remoto. En ese entonces soportaban Android, Iphone y Blackberry. Revelaciones similares se realizaron en 2013 y 2014.[6] [7]

1.2 Vigilancia Estatal en América Latina

No es necesario tener las capacidades de vigilancia de gobiernos como Estados Unidos para vigilar a los ciudadanos de un país. Es suficiente tener los recursos para comprar las soluciones ofertadas por las empresas que venden tecnologías de vigilancia. En los últimos años sucedieron revelaciones que muestran que gobiernos de América Latina compraron software de vigilancia y que lo han utilizado para espiar a periodistas, activistas y políticos.

En julio de 2015, la empresa italiana Hacking Team fue vulnerada electrónicamente y se hicieron públicos 400 gigabytes de información. Un informe realizado por la organización chilena Derechos Digitales basada en la información filtrada muestra que Brasil, Chile, Colombia, Ecuador, Honduras, México y Panamá compraron licencias de software de esta empresa; mientras que Argentina, Guatemala, Paraguay, Uruguay y Venezuela mantuvieron negociaciones que, hasta julio de 2015, no se concretaron. [8]

Un video de Hacking Team filtrado a Wikileaks describe el tipo de software que ellos ofertaban a gobiernos en 2011. El mismo dice tener la capacidad de infectar teléfonos y computadoras de forma remota o con acceso físico. Una vez que el dispositivo está infectado, se tiene control remoto sobre el mismo y se puede vigilar su actividad. Comunicaciones cifradas, documentos editados, impresiones, capturas de actividad en el teclado (*keylogger*), mensajería instantánea son algunas de las actividades que se puede monitorear de forma remota.[9] En otro documento promocional de 2011 se dice que tiene la capacidad de monitorear desde unas pocas personas hasta algunos miles. [10] Basado en información de las filtraciones, el informe de Derechos Digitales dice que en México, Panamá y Ecuador se utilizó el software Hacking Team para espionaje político.

En la imagen 3 se puede ver un gráfico que detalla cuánto gastaron los países de América Latina en el software de Hacking Team. El medio periodístico Animal Político de México hizo otra investigación sobre los documentos y determinó que México es el país que más dinero pagó a Hacking Team a nivel mundial y Chile ocupa el quinto lugar.[11]

NSO es una empresa Israelita que vende el software Pegasus, que de manera similar al de Hacking Team, permite tomar control remoto de teléfonos celulares. En una investigación realizada por el laboratorio Citizen Lab de la Universidad de Toronto sobre el uso de Pegasus a nivel mundial mostró que México es el país que más utiliza este software.[12]

Las organizaciones R3D, Article 19 y SocialTIC realizaron el reporte “Gobierno Espía” que analiza el uso de este software en México. El reporte mostró intentos de infectar teléfonos celulares de periodistas y activistas políticos que denunciaron casos de violencia y corrupción en ese país. Entre las víctimas se encuentran miembros del Centro de Derechos Humanos Miguel Agustín Pro Juárez que denunciaron el caso de los 43 estudiantes desaparecidos de Ayotzinapa.[13]

Imagen 3: Hacking Team en América Latina
Fuente: Derechos Digitales

La periodista Carmen Aristegui denunció el caso de corrupción Casa Blanca que vincula al presidente mexicano Enrique Peña Nieto en su programa Aristegui Noticias. Luego de esto, ella fue otra de las víctimas de intentos de infección del software de NSO. Al no poder vulnerar el teléfono de la periodista intentaron infectar el teléfono de su hijo, que en ese entonces era menor de edad. Esto muestra que no hay que ser político, periodista o activista para ser víctima de espionaje; basta con ser hijo o cercano a alguien a quién un estado quiere vigilar para que las comunicaciones puedan ser intervenidas.

La empresa alemana Fin Fisher vende soluciones similares a las de Hacking Team y NSO. Para su funcionamiento necesita tener servidores en

Internet a los que reporta la actividad de los dispositivos infectados. Citizen Lab encontró cómo identificar estos servidores al enviar cierto tipo de peticiones. Para eso utilizaron herramientas como ZMAP[2] para explorar todas las direcciones públicas IPV4 de Internet. Con esto identificaron 33 posibles clientes ubicados en 32 países; en el listado aparecen México, Venezuela y Paraguay. Debido a las limitaciones del estudio no se sabe exactamente quiénes están utilizando el software de Fin Fisher para espiar, pero sí que se lo ha utilizado en estos países.[14]

Otro caso que puso en evidencia Citizen Lab es el denominado PACRAT. En el mismo se identificó infraestructura compartida por varios países de la región, así como campañas para intentar infectar con código malicioso computadoras o robar contraseñas a través de *phishing*[3]. Entre los países que aparecen en este estudio se encuentran Argentina, Ecuador, Brasil y Venezuela. En este caso el espionaje se dio a políticos, periodistas y activistas. [15]

Casos como los de Hacking Team, NSO, Fin Fisher y PACRAT muestran que gobiernos en América Latina adquieren tecnología de vigilancia. La misma que debería ser utilizada para combatir el crimen se utiliza también para espiar a políticos, periodistas o activistas. Esto se hace incluso al margen de la ley.[8] En los cuatro casos se trata de vigilancia dirigida; es decir que se quiere espiar a ciertas personas en particular.

La vigilancia masiva es cuando se espía o se recolecta datos de forma masiva a una población determinada. La Fundación de la Frontera Electrónica (EFF por sus siglas en inglés) coordinó una investigación en 12 países de América Latina para conocer las capacidades de vigilancia de los gobiernos de estos países.[16]

De los países en los que se realizó la investigación, se determinó que en México, Honduras, Colombia, Perú, Argentina y Chile existe obligación legal de las operadoras de telecomunicaciones de recolectar

2 ZMAP es una herramienta que permite escanear todas las direcciones IPv4 públicas. Disponible en: https://zmap.io/

3 Ataque informático que intenta robar claves a través de páginas falsas similares a las reales.

datos de sus abonados. Esta obligación va desde al menos un año en Chile hasta los 5 años en Colombia y Honduras.

En todos los casos para que el gobierno pueda acceder a la información recolectada se requiere una orden judicial. La excepción a la regla es México donde el gobierno puede acceder a los datos de localización en tiempo real.

La organización Privacy International realizó en 2015 una investigación sobre la vigilancia en Colombia. En la misma se muestra que en 2013, la policía colombiana gastó 28 millones de dólares estadounidenses para el fortalecimiento de la Plataforma Única de Monitoreo y Análisis (PUMA). Más de la mitad de esta inversión se utilizó para:

> … software y hardware básicos necesarios para convertir PUMA en un sistema completo de interceptación legal, capaz de recopilar datos y contenido de llamadas de voz, VoIP, tráfico de Internet y redes sociales en 12 de los proveedores de servicios de telecomunicaciones de Colombia – cuatro redes de datos móviles y voz (Claro, Tigo, Avantel y Movistar) y ocho proveedores de servicios de Internet (Une, Telefónica, Emcali, Metrotel, ETB, Telebucaramanga, Telmex y EPM). [17]

Este contrato fue firmado con la empresa Verint aunque luego se canceló; sin embargo, dice el informe, el proyecto PUMA sigue activo.

El objetivo de este trabajo no es detallar todos los casos de vigilancia por parte de gobiernos en América Latina. Sin embargo, es importante saber que los gobiernos de la región tienen la capacidad de vigilar a sus ciudadanos y que en varias ocasiones abusaron de ese poder, incluso al margen de la ley.

1.3 Modelo de Amenaza

En las secciones previas de este capítulo se vio que existen varias formas en la que los gobiernos y empresas pueden espiar las comunicaciones en Internet. Con la vigilancia masiva se intenta espiar o

recolectar la información de poblaciones enteras y con la vigilancia dirigida se busca espiar a un grupo específico de gente.

El modelo de amenaza que busca mitigar este trabajo es el de vigilancia masiva. Se quiere proteger el tráfico de la comunicación de posibles actores que podrían espiarlo. Estos podrían ser los proveedores de Internet o cualquier otro que pueda interceptar las comunicaciones en tránsito. Además se quiere proteger las comunicaciones de las personas u organizaciones que proveen los servicios de comunicación como correo electrónico, chat y voz sobre IP.

La imagen 4 explica el modelo de amenaza para el caso de correo electrónico; pero es un esquema similar a cualquier tipo de servicio. La conexión a Internet de Alice o Bob podría ser espiada por su proveedor local de Internet o por cualquier ruteador en el medio hasta llegar al servidor de correo. El proveedor de correo electrónico podría espiar los correos de todos sus usuarios.

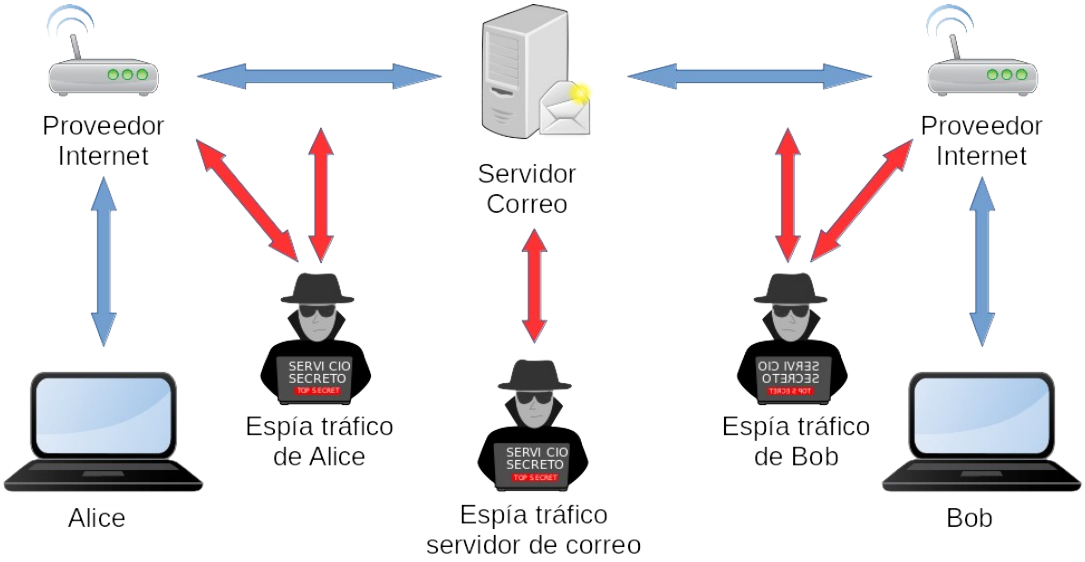

Imagen 4: Modelo de amenaza
Fuente: Elaboración propia con gráficos de Openclipart

En este trabajo no se analiza el modelo de amenaza de atacantes como NSO, Fin Fisher o Hacking Team. Es importante tomar en cuenta la seguridad de las terminales de comunicación debido a que si alguna estuviera infectada con software malicioso pondría en riesgo la comunicación.

2 Privacidad, Software Libre, Criptografía y Autonomía

Eben Moglen es un abogado y doctor en historia de la universidad de Yale que financió sus estudios desarrollando software. Fue miembro del equipo de defensa de Philip Zimmermann, autor original de PGP[4], cuando fue amenazado legalmente por el gobierno de Estados Unidos por hacer accesible el cifrado fuerte libremente. Ha sido abogado de la Fundación de Software Libre, organización con la que trabajó en el desarrollo de la versión tres de la licencia GPL. Desde 1987 trabaja como profesor en la Universidad de Columbia en Estados Unidos. [18] [19]

Moglen tiene un entendimiento técnico y político sobre el uso de la tecnología y el impacto que esta puede tener en la sociedad. Luego de las primeras revelaciones de Snowden, Moglen dictó una seria de charlas en la Universidad de Columbia. Él sostiene que para tener privacidad en Internet es necesario tener por lo menos 3 cosas: secreto, anonimato y autonomía. [20]

El secreto es la habilidad de ocultar el contenido de los mensajes para que pueda ser leído solo por quiénes intervienen en la comunicación. El anonimato por su parte implica ocultar quién publica contenido y quién lo lee, incluso si el contenido del mensaje es público. Dice Moglen que el anonimato de quién publica algo es tan importante como el de quién lo lee. Autonomía quiere decir que se puede tomar decisiones propias independientemente de las fuerzas que quieran vulnerar el secreto o el anonimato.

Secreto de las comunicaciones, anonimato y autonomía son los pilares que se buscarán al momento de elegir una herramienta de seguridad en este trabajo.

En la sección 2.1 se explicará que es la privacidad y por que es importante protegerla. En la sección 2.2 se hablará sobre software libre y las

4 Software que permite cifrar correos electrónicos sobre el que se hablará en el capítulo 4.

características que lo hacen importante para las comunicaciones seguras. En la sección 2.3 se describe el uso de la criptografía para proteger las comunicaciones. Por último, en la sección 2.4 se analiza la centralización de las comunicaciones en Internet y se explica por qué es importante tener autonomía.

2.1 Privacidad

Katitza Rodríguez, miembro de la EFF, en un trabajo realizado en conjunto con la Fundación Acceso de Centro América explica que la privacidad puede ser entendida de dos formas. La primera desde el punto de vista del derecho a ser dejado solo, es decir nadie debe entrometerse en la vida privada de las personas. La segunda que considera que la privacidad es el derecho de las personas a escoger de que manera y en qué circunstancias exponer su comportamiento a los demás. [21]

En el mundo físico es fácil entender este concepto. Si Alice y Bob se encuentran en un parque para conversar y no lo comentan con nadie, es muy probable que nadie sepa que esa reunión existió y menos aún sobre que conversaron.

Desde el punto de vista de ser dejados solos, la privacidad de Alice y Bob es respetada salvo que se estuviera espiando, fotografiando o grabando la conversación. Por otro lado, Alice y Bob tienen la capacidad de decidir con quién compartir el hecho de que la reunión sucedió y el contenido de la misma.

En el mundo digital sucede lo contrario, toda comunicación deja registro salvo que se haga algo al respecto. El proveedor de Internet o alguien que trabaje en el mismo puede vigilar el tráfico de sus usuarios e incluso guardar registros del mismo. El proveedor de nombres de dominio (DNS por sus siglas en inglés) puede saber todos los dominios al que quiere acceder una determinada dirección IP. Los portales web o servidores de servicios en línea pueden conocer la actividad de sus usuarios. En muchos

casos los Estados pueden vigilar el tráfico de los ISPs o proveedores en línea de manera legal o ilegal como se vio en el capítulo anterior.

La privacidad es un derecho humano fundamental consagrado en la Declaración Universal de Derechos Humanos donde en su artículo 12 dice que:

> Nadie será objeto de injerencias arbitrarias en su vida privada, su familia, su domicilio o su correspondencia, ni de ataques a su honra o a su reputación. Toda persona tiene derecho a la protección de la ley contra tales injerencias o ataques.[22]

El derecho a la privacidad o intimidad se encuentra protegida en la constitución y leyes de varios países. En el caso de Ecuador la no interferencia de las comunicaciones privadas esta protegida en el artículo 66 literal 21 de la constitución:[23]

> El derecho a la inviolabilidad y al secreto de la correspondencia física y virtual; esta no podrá ser retenida, abierta ni examinada, excepto en los casos previstos en la ley, previa intervención judicial y con la obligación de guardar el secreto de los asuntos ajenos al hecho que motive su examen. Este derecho protege cualquier otro tipo o forma de comunicación

Países como Nicaragua, Honduras, El Salvador y Guatemala también lo hacen. A pesar de que en estos países las constituciones protegen la privacidad de los ciudadanos, existen leyes que la pueden contradecir. [21] [24]

Existen marcos legales que protegen derechos como la privacidad incluso a nivel constitucional; sin embargo no existe garantía de que siempre se cumpla la ley. De manera similar a como se ponen cerraduras en las puertas de las casas, se debe buscar soluciones tecnológicas para proteger las comunicaciones en línea y no depender solamente de la buena voluntad del Estado.

En un mundo donde toda actividad deja huella, buscar comunicaciones secretas no es algo subversivo, es ejercer el derecho a la privacidad.

2.2 Software Libre

Software libre son los programas informáticos donde “los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.”[25] Software no libre o privativo es el que no cumple con estas libertades. Por ejemplo, con el software no libre se puede limitar estudiar como funciona un sistema o prohibir que se lo comparta.

Los sistemas operativos más utilizados en computadoras personales o dispositivos móviles no son libres. El código fuente de Microsoft es cerrado y no es accesible al público. Los sistemas operativos de Apple, si bien tienen código proveniente de sistemas libres como FreeBSD[26] son sistemas cerrados. Incluso el sistema operativo Android, que en su mayoría es software libre, [27] suele tener dependencias de herramientas cerradas de Google como Google Maps, Google Play u otras[5].

Las tres empresas desarrolladoras de estos sistemas operativos han participado en el programa de vigilancia PRISM de la NSA. La situación es más compleja porque a más de tener el control sobre los sistemas operativos de las computadoras personales y dispositivos móviles proveen servicios en la nube; al igual que Facebook. En este caso a más de ceder el control del software a una empresa se cede el control de los datos.

Si se toma como ejemplo al servicio de correo electrónico Gmail, de Google, es conocido que hasta junio del año 2017 los anuncios publicitarios de Gmail eran asociados al contenido de los correos. Es decir que Google estaba analizando los correos de todos sus usuarios para proveer anuncios personalizados. El 23 de junio de 2017 Google anunció que ya no mostraría anuncios personalizados junto a los correos de Gmail.[28] Qué no se muestre anuncios junto a los correos no es garantía de que no se este procesando el contenido de los mismos.

En agosto de 2018 la agencia de noticias de Estados Unidos Associated Press realizó una investigación donde demostró que las

5 El sistema operativo LineageOS se basa en Android y por defecto no instala las aplicaciones cerradas de Google. https://lineageos.org

aplicaciones de Google para IOS y Android almacenan la localización de sus usuarios incluso cuando los mismos desactivan esta funcionalidad.[29] A finales del mismo mes Bloomberg publicó un artículo donde denunció un acuerdo secreto entre Google y Mastercard para verificar si las compras realizadas con tarjetas de crédito fuera de línea se realizaban gracias a un anuncio de Google.[30] ¿Se puede confiar en Google para defender la privacidad? ¿Se lo puede hacer con las otras empresas de PRISM?

Richard Stallman, fundador del movimiento de software libre, explica que el software privativo como el software como servicio tienen el problema de ceder poder a otro. En el caso del software privativo se cede el control del software a el desarrollador o la empresa que hace el sistema. En el caso del software como servicio, además se cede el control sobre los datos al proveedor del servicio.[31]

Explica Stallman que “[a] diferencia del software privativo, el servicio sustitutivo del software no requiere código oculto para obtener los datos de los usuarios. Son los usuarios quienes tienen que enviar sus propios datos al servidor para poder usarlo. Esto tiene el mismo efecto que el *spyware*: el administrador del servidor obtiene los datos, sin ningún tipo de esfuerzo, en virtud de la naturaleza misma del servicio sustitutivo del software.” [32]

Con software libre se puede estudiar como funciona el sistema, mejorarlo y contribuir a su desarrollo. Un proyecto exitoso de software libre es aquel que forma comunidad. En el año 2017, el núcleo Linux, se desarrolló por 15600 personas de más de 1400 empresas alrededor del mundo.[33] El hecho de tener una comunidad grande de desarrolladores no es garantía de que el sistema sea seguro, pero sí de que no se depende de una sola empresa para su seguridad.

No todo proyecto de software libre es exitoso ni tiene una comunidad tan grande. En general antes de adoptar un sistema libre es importante informarse sobre el tamaño y la calidad de su comunidad.

Que el código fuente sea público permite que el mismo pueda ser auditado por gente de todo el mundo. Esto por si solo no garantiza de que el

sistema sea auditado pero a diferencia del software privativo tiene un modelo de desarrollo transparente. Si la comunidad es grande es probable que alguien vigile el código fuente, pero tampoco es garantía que esto suceda. Por esto es importante que desde la academia, gobierno y sociedad civil se audite a los sistemas libres que brinden seguridad en las comunicaciones.

Otro factor a destacar sobre el software libre es que no existe una barrera económica para acceder al mismo. La libertad de poder distribuir el software permite que ricos y pobres puedan tener acceso a las mismas herramientas para comunicarse de forma segura.

El software libre por si solo no es garantía de que las comunicaciones sean seguras. Para que esto suceda es indispensable el uso de la criptografía en las comunicaciones.

2.3 Criptografía

Luego de las revelaciones de la NSA se le preguntó a Snowden si la criptografía funciona, a lo que respondió: “La criptografía funciona. Los sistemas criptográficos correctamente implementados son una de las pocas cosas en las que podemos confiar. Lamentablemente la seguridad de las terminales es tan débil que la NSA puede encontrar sus caminos”[6][34]

A más de las declaraciones de Snowden, cabe destacar que documentos revelados muestran que en 2012 un correo cifrado con PGP[7][35] y una conversación de chat cifrada con OTR[8][36] no pudieron ser leídas por la NSA.

La criptografía puede tener varios tipos de usos para mantener el secreto de la información. Se la puede utilizar para cifrar las comunicaciones en tránsito; se puede cifrar la comunicación extremo a extremo; o se la puede utilizar para cifrar la información de datos almacenados.[37]

6 Traducción propia del inglés “Encryption works. Properly implemented strong crypto systems are one of the few things that you can rely on. Unfortunately, endpoint security is so terrifically weak that NSA can frequently find ways around it. ”

7 PGP quiere decir “*Pretty Good Privacy*”, la traducción al español sería “Muy Buena Privacidad” En el capítulo 4 se verán los aspectos técnicos de este protocolo.

8 Sistema de cifrado de chat que se explicará en el capítulo 5.

El cifrado de datos almacenados sirve para proteger la información que se encuentra en un dispositivo. Casos típicos de uso son el cifrado de discos duros, otros dispositivos de almacenamiento o archivos. De esta manera si alguien roba un equipo no podrá acceder a la información del mismo. En este trabajo no se abordará este tipo de cifrado pero se recomienda tenerlo en cuenta; particularmente por si un dispositivo llega a ser robado.

El cifrado de la comunicación en tránsito se utiliza para proteger la comunicación entre una aplicación cliente y un servidor. Para acceder a la banca en línea, por ejemplo, se utiliza el protocolo HTTPS. El mismo sirve para proteger la comunicación entre el navegador web (cliente) y el sistema del banco.

El cifrado extremo a extremo da un paso más allá. Si Alice envía un mensaje a Bob a través de un servidor administrado por Eva, este mensaje podría ser visto por la administradora. Para que esto no suceda, el mensaje que Alice envía a Bob sale cifrado de la máquina de Alice y se descifra en la de Bob. Nadie en el camino podrá leer este mensaje, incluso quien administra el servidor.

*****

Adicionalmente al secreto de las comunicaciones, existen otras características de la criptografía que son deseables en las comunicaciones secretas como: autenticación, repudio y secreto perfecto hacia adelante[9].

La autenticación consiste en que si Alice envía un mensaje a Bob, ella debe tener certeza que se está comunicando con Bob y no con alguien más. La autenticación también permite evitar el riesgo del ataque de hombre en el medio.

A diferencia del mundo corporativo en las conversaciones secretas se desea tener repudio. Es decir que si alguien tiene acceso a un mensaje

9 Traducción tomada de Wikipedia, en inglés se conoce como “perfect forward secrecy”

escrito por Alice, este mensaje no será una evidencia criptográfica de que Alice fue quien lo envió.

El secreto perfecto hacia adelante es cuando cada mensaje está cifrado con una clave diferente por lo que no será posible descifrar todos los mensajes recuperando la llave privada de Alice. Para lograr esta propiedad, en lugar de cifrar los mensajes con la llave pública se negocia una llave simétrica efímera con algoritmos como Diffie Hellman. [38]

2.3.1 Criptografía de Ayer, Hoy y Mañana

Snowden sostiene que la criptografía es algo en lo que se puede confiar, no existe garantía de que esto vaya a ser así siempre. En los años 90s existió un conflicto entre el gobierno de los Estados Unidos y la comunidad activista conocida como *Cypherpunks* respecto al uso de la criptografía. Hasta entonces el gobierno de Estados Unidos, representado por la NSA, consideraba que la criptografía era un arma estratégica de guerra y su uso debía ser limitado. Los *Cypherpunks* consideraban que la criptografía era una herramienta que permitía a las personas comunicarse con libertad y preservar la privacidad.[39]

Un caso emblemático fue el del sistema de cifrado de correo conocido como PGP desarrollado por Phil Zimmermann en el año 1991. Zimmermann creó este sistema porque hasta ese entonces la criptografía había sido utilizada solamente por gobiernos, diplomáticos y militares; él creía que "… una emergente economía global dependiente cada vez más de las comunicaciones digitales, las personas comunes y las empresas necesitan criptografía para proteger sus comunicaciones diarias."[10] [40]

El software y el código fuente del sistema fueron publicados en Internet sin costo en junio de 1991 y se distribuyó rápidamente por varios países; esto trajo problemas a Zimmermann. Por un lado violaba el uso de

10 Traducción propia del inglés: "... an emerging global economy depending more and more on digital communication, ordinary people and companies need cryptography to protect their everyday communications."

patente del algoritmo RSA[11] que pertenecía a la empresa RSA Data Security; por otro lado molestó al gobierno ya que violaba las leyes de exportación al considerar la criptografía como un arma estratégica. Zimmermann fue investigado por democratizar el acceso a la criptografía; a pesar de esto nunca fue culpado.[39] [41]

A finales de los años 1970s el gobierno federal de Estados Unidos publicó el estándar DES que fue ampliamente utilizado. Durante los años 1990s se cuestionó la seguridad de este estándar, algo que el gobierno negaba. La EFF, con un presupuesto de $250 000 desarrolló hardware y software capaces de romper el algoritmo DES en 56 horas. Hizo esto para demostrar que el algoritmo no era seguro y no se podía confiar en el mismo. [42]

Los documentos de Snowden muestran que la NSA hace lo posible para que no existan comunicaciones que ellos no puedan espiar. Una investigación realizada por la revista alemana Der Spiegel, basada en documentos de la NSA, puso en evidencia que la agencia tiene proyectos dedicados a vulnerar las seguridades en Internet. En esa investigación se destacan ataques a tecnologías de VPN como PPTP que la NSA puede vulnerar fácilmente; también se muestran intentos de la agencia de influir en los organismos de estándares como IETF[12] o NIST[13][43]

La agencia Routers denunció que la NSA habría dado diez millones de dólares a la firma RSA para que utilizara el algoritmo de Curvas Elítpicas Dual como generador de números aleatorios en su producto Bsafe. Este algoritmo ha demostrado ser débil por lo que el experto Bruce Schneier sostiene que se trata de una puerta trasera.[44]

La investigación de Der Spiegel también dice que herramientas como Tor, PGP u OTR no han podido ser vulneradas por lo NSA. Eso se sabe al menos hasta dichas revelaciones que sucedieron en el año 2013.

11 Algoritmo de cifrado asimétrico.
12 Grupo de Trabajo de Ingeniería de Internet – IETF por sus siglas en inglés.
13 Instituto Nacional de Estándares y Tecnología – NIST por sus siglas en inglés.

¿Será posible que ahora lo puedan hacer? ¿Es posible que lo pueden hacer en el futuro?

El problema se agrava con la computación cuántica que probablemente haga vulnerable las técnicas de cifrado que se utilizan hoy en día. Una investigación realizada por The Washington Post y sustentada por los documentos de Snowden denuncia que la NSA invierte presupuesto para computación cuántica con el fin de vulnerar la seguridad de los sistemas que se utilizan en la actualidad.[45]

La NSA y probablemente agencias de inteligencia de otras potencias intenten vulnerar las herramientas de criptografía que se utilizan en estos días. Por esto es importante que en América Latina se generen capacidades técnicas para entender y proponer criptografía post cuántica. Es decir el tipo de cifrado que podría resistir a la computación cuántica.

En ese sentido es importante destacar el trabajo que realiza la comunidad académica de Criptografía Post Cuántica. Pedro Hecht, coordinador académico de la Maestría de Seguridad Informática de la Universidad de Buenos Aires, ha desarrollado algoritmos que podrían resistir a la computación cuántica y que permitirían que en el futuro se pueda tener privacidad en las comunicaciones. [14]

Probablemente para América Latina sea difícil tener computación cuántica; sin embargo la criptografía post cuántica permite igualar la asimetría de poder entre quién puede tener computadoras cuánticas y quiénes necesitan proteger su vida privada.

2.4 Autonomía en las Comunicaciones

Julian Assange dice que: “Hay muchos aspectos de Internet que no están suficientemente descentralizados, como su infraestructura física, por ejemplo. Eso hace que sea más vulnerable a la vigilancia masiva...”. Sobre la región Assange dice que “[e]n América Latina, casi todas las conexiones a

14 Los trabajos de Pedro Hecht están disponibles acá: https://arxiv.org/a/hecht_p_1.html

la Internet mundial pasan a través de cables de fibra óptica que atraviesan Estados Unidos”[46]

Servicios de correo como Gmail, Hotmail o Yahoo; servicios de chat como Whatsapp, Facebook Messenger; servicios de voz sobre IP como Skype o Google Hangouts; tienen todos en común que son provistos por empresas participantes del programa PRISM.

En el caso de chat o de voz, estos son servicios centralizados y no pueden federarse. Si Alice quiere hablar con Bob a través de Whatsapp, los dos deben tener una cuenta en Whatsapp. No es posible que Alice hable con Bob desde su cuenta de Telegram. Lo mismo sucede con servicios de voz sobre IP como Skype.

En el caso de correo electrónico es diferente porque este es un sistema federado. Alice puede utilizar la cuenta del servidor de su organización para comunicarse con la cuenta de correo de Bob en Gmail. Si bien el correo electrónico es federado, también existe concentración de cuentas en servicios como Gmail, Outlook y Yahoo.

Para tener comunicaciones seguras es importante tener autonomía El software libre da la posibilidad de que cualquier organización con las suficientes capacidades técnicas pueda implementar sus propios servidores de comunicaciones para que un tercero no las espíe. Los servicios ocultos de redes de anonimato como I2P y Tor permiten tener infraestructura propia de comunicaciones sin la necesidad de dominios, direcciones IP públicas y de manera anónima.

3 Redes de Anonimato y VPNs

La primera medida que debería tomarse para conseguir comunicaciones secretas es el anonimato, es decir separar la actividad en línea de la identidad de la persona. En el caso de este trabajo se busca tener anonimato respecto a cualquiera que pueda interceptar las comunicaciones, pero no entre los interlocutores. Es decir Alice quiere saber que está hablando con Bob y viceversa pero nadie más tiene que saber que la conversación sucedió. Existen varias herramientas que permiten tener anonimato, en este trabajo se verán tres.

La primera es a través de redes virtuales privadas conocidas como VPNs por sus siglas en inglés. El uso de VPNs para el contexto de este trabajo consiste en establecer una conexión cifrada a un servidor en otra ubicación geográfica y enviar todo el tráfico de Internet por esa conexión.

La segunda opción es utilizar una red de anonimato como Tor y acceder a servicios públicos sin revelar la identidad real ni la dirección IP. De esta forma Alice podría crear cuentas de correo electrónico o de chat de manera anónima.

La tercera es evitar la necesidad de confiar en un servidor público. En este caso se utilizan los servicios ocultos provistos por redes de anonimato como Tor o I2P. Así se puede publicar servicios de Internet de forma anónima y sin necesidad de direcciones de IP públicas.

Los sistemas o redes provistas a través de servicios ocultos se los conoce como redes obscuras[15] dentro de la comunidad de anonimato en Internet, mientras que a la Internet convencional se la conoce como la red clara[16]. Una diferencia fundamental entre las redes Tor e I2P es que la primera esta pensada para el anonimato en la red clara y el soporte de redes obscuras, mientras que la segunda esta pensada para crear redes obscuras.

15 Traducción propia del inglés “*dark web*”
16 Traducción propia del inglés “*clear web*”

En la sección 3.1 se explicará el uso de las VPNs; en la sección 3.2 la red de anonimato Tor; en la sección 3.3 la red I2P; por último se termina el capítulo en la sección 3.4 con un análisis sobre las herramientas presentadas.

3.1 Redes Virtuales Privadas - VPNs

Las VPNs permiten establecer canales de comunicación cifrados a través de Internet. Tradicionalmente son utilizadas para acceder a una red corporativa desde Internet de forma segura. Existen también proveedores de VPN que permiten enviar el tráfico de Internet a través de ellos y de esa manera poder protegerse de un ataque a través de una red wifi o la red local. [47]

En la imagen 5 se puede ver la descripción de una VPN típica. En este caso Alice establece un canal cifrado hasta el proveedor de VPN por donde enviará todo su tráfico para acceder a los servidores de chat, correo y web. El proveedor de Internet o un espía que intercepte la conexión a Internet de Alice no podrá saber el contenido de la comunicación; tan solo verá tráfico desde la computadora de Alice hasta una VPN. Los administradores de los servicios de correo, web, chat o cualquier otro no sabrán desde donde se conecta Alice ya que ella está usando la dirección IP de la VPN.

El proveedor VPN tendrá el mismo acceso a las comunicaciones de Alice que tendría el proveedor local de Internet cuando no se usa una VPN. Es decir que tiene la posibilidad de espiar el historial de navegación, servicios accedidos, etcétera. En caso de utilizar un servicio de VPN es importante seleccionarlo con cuidado ya que este podría monitorear el tráfico.

Incluso si el proveedor de VPN tiene voluntad de defender la privacidad de sus usuarios, este podría ser presionado por un Estado u otro adversario a entregar los registros de acceso de Alice. Es importante notar que quienes espían el tráfico de Alice de manera local no podrán monitorear

su actividad; los proveedores de servicio no sabrán desde donde se conecta. A pesar de esto, en los dos casos es posible identificar al proveedor de servicio VPN y de esta manera presionarlo para que entregue la información de Alice.

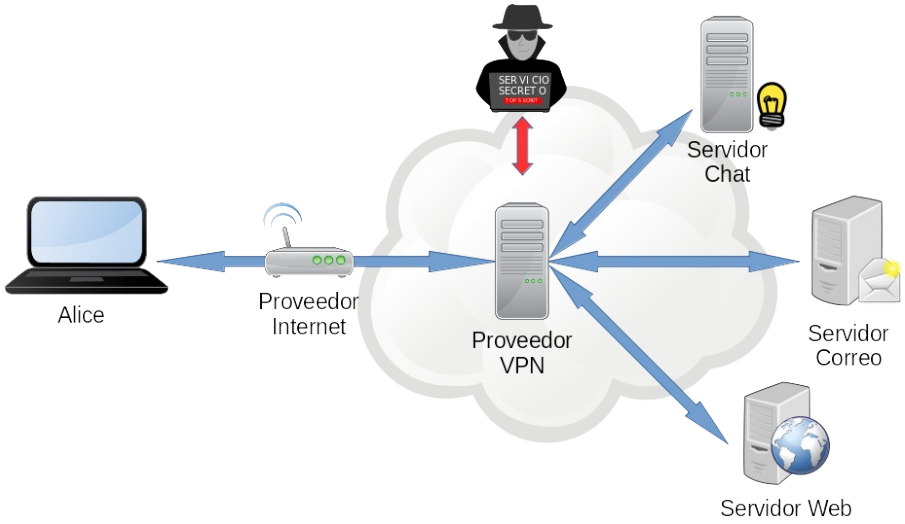

Imagen 5: Red Privada Virtual - VPN
Fuente: Elaboración propia con gráficos de Openclipart

Las VPNs son herramientas útiles para las comunicaciones seguras, pero se debe depositar demasiada confianza en el proveedor; para el propósito de este trabajo se busca tener un anonimato más fuerte.

3.2 La Red Tor

La red Tor permite anonimizar el tráfico de servicios que funcionan con el protocolo TCP. En esencia se puede utilizar Tor para ocultar la IP de origen del tráfico de navegación web, correo electrónico, transferencia de archivos, chat, entre otros. En el caso de navegación, por ejemplo, el proveedor de Internet o administrador de la red solo puede saber que el usuario está utilizando Tor, pero no para que. El servidor web al que accede el usuario no conoce la dirección IP de origen; sino que verá una dirección IP de un nodo de la red Tor que probablemente estará en otro país.

La técnica utilizada por Tor para conseguir anonimato se conoce como enrutamiento cebolla[17]. Si Alice quiere acceder al servicio provisto por

17 Conocido como *onion routing* en inglés.

Bob debe establecer comunicación con tres nodos. El primero conoce la IP origen de Alice y la IP del segundo; el segundo sabe la IP del primero y la del tercero; el tercero conoce la IP del segundo y la IP de Bob, es decir del destino como se puede ver en la imagen 6.[48]

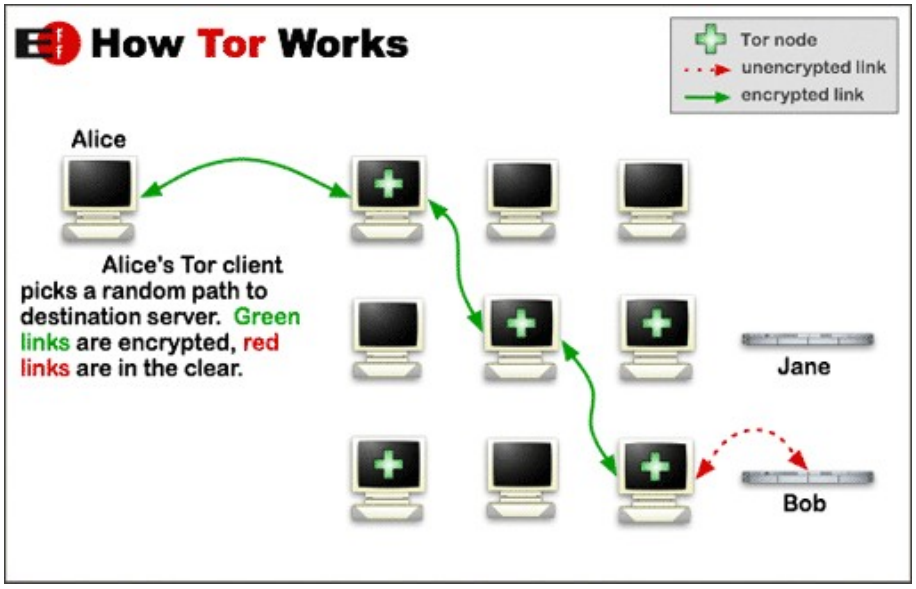

Imagen 6: Ruteo cebolla
Fuente: Electronic Frontier Foundation

La lista de servidores se descarga de un grupo de 9 nodos conocidos como autoridad de directorio, los mismos mantienen un listado verificado de todos los nodos disponibles en la red y sus llaves públicas. Esta información es replicada a otros 100 servidores. El listado de los 9 nodos y sus réplicas se encuentra almacenado en el código fuente de Tor y es lo que se usa para identificar los tres nodos que forman los circuitos de conexión.[49]

En febrero de 2018 existían 9 servidores de autoridad de directorio.[50] Durante el año 2017 la red estaba compuesta en promedio por 6984 nodos encargados de rutear el tráfico de 2 336 061 personas en promedio. Se llegó a un máximo de 4 238 515 el 28 de diciembre de 2017.[51]

En el cliente funciona un programa llamado Onion Proxy (OP) que implementa el protocolo SOCKS. Este protocolo funciona como intermediario del tráfico de TCP y UDP para que poder establecer comunicaciones incluso

detrás de cortafuegos.[52] Tor puede ocultar el tráfico de cualquier protocolo TCP compatible con SOCKS.

La comunicación entre el cliente y el servidor se la hace a través de TLS[18] mediante claves públicas de larga duración para mantener la identidad de los nodos de la red conocidos como Onion Routers (OR). Los OP no utilizan llaves públicas ya que no quieren ser identificados. La comunicación entre OP con OR y de OR con OR es protegida a través de cifrado simétrico con AES. Se utiliza Diffie Hellman para el intercambio de clave de sesión que además sirve para tener secreto perfecto hacia adelante. [53]

La comunicación funciona de la siguiente forma:

1. Se establece un canal cifrado con TLS entre el OP y el primer OR del circuito.

2. Se establece un canal TLS entre el OP y el segundo OR ruteando el tráfico a través del primer OR.

3. A través del primer OR y el segundo OR se establece conexión TLS con el tercer OR.

4. El circuito de Tor queda formado y se puede utilizar para navegar en Internet de forma anónima o para cualquier otro servicio TCP.

Al tercer OR se lo conoce también como nodo de salida. Hay que prestar atención al mismo ya que si bien la comunicación va cifrada dentro de la red Tor, no hay garantía que esta este cifrada a la salida. Si el servidor web que está ejecutando Bob utiliza HTTP en lugar de HTTPS, entonces el nodo de salida podría espiar el contenido de la comunicación.

3.2.1 Acceso a la Red Tor

Para acceder la red Tor se debe instalar el proxy Tor en el sistema y configurar la aplicación cliente para que utilice el proxy SOCKS por el puerto 9050. Con el fin de facilitar esta configuración el proyecto Tor ha desarrollado el Navegador Tor.

18 Seguridad de Capa de Transporte – TLS por sus siglas en inglés.

Esta aplicación es una versión modificada de Mozilla Firefox con mejoras de seguridad y privacidad. Trae instalado complementos como HTTPS Everywhere que intenta conectarse siempre a sitios HTTPS para evitar posible espionaje de los nodos de salida; trae NoSript que permite bloquear funcionalidades de Javascript que podrían quitar el anonimato al usuario; además ejecuta el proxy Tor en el puerto 9150 por lo que otras aplicaciones también se podrían conectar por ahí.

La instalación es simple, basta con descargar el binario y ejecutarlo. El resultado es un navegador que permite acceder a sitios web de forma anónima. Adicionalmente tiene opciones para configurar Tor y ver los países por donde atraviesa el tráfico como se puede ver en la imagen 7.

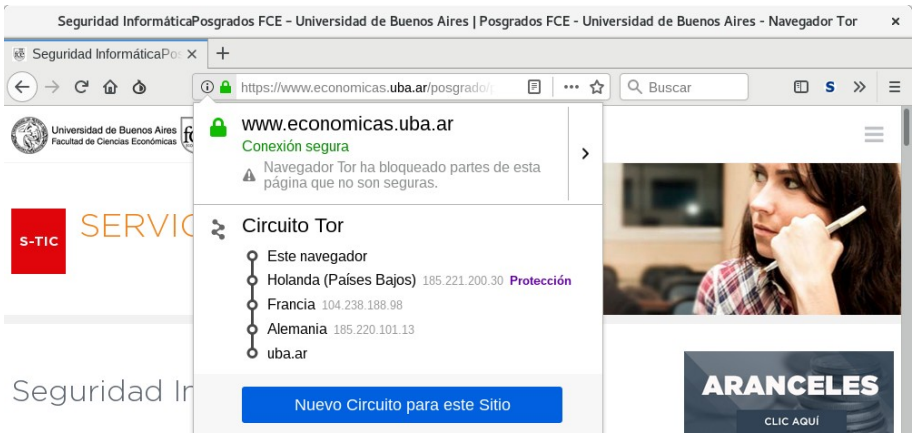

Imagen 7: Navegador Tor
Fuente: captura de pantalla

La instalación es simple pero su uso requiere cambio de hábitos. El tiempo de respuesta de las páginas es más lento y algunas bloquean su tráfico obligando al usuario a resolver *captchas*.

Se puede acceder a la red Tor a través de sistemas Android utilizando la aplicación Orbot. Esta aplicación tiene funcionalidad tipo VPN que permite seleccionar otras aplicaciones para que salgan a través de Tor, por ejemplo un navegador de Internet o un cliente de correo. En el caso de IOS existe la aplicación Onion Browser que permite navegar a través de Tor.

Puede suceder que la conexión a la red Tor esté bloqueada con lo que no se podría acceder a la misma. Para solucionar este problema, se pueden utilizar los nodos *bridges* que son nodos de acceso que no se

publican junto a los demás y se los solicita de manera distinta. Tanto el navegador Tor como Orbot permiten configurar los *bridges* para acceder a la red cuando se depende de conexiones censuradas. Otra opción para evitar la censura de Tor es conectarse a la red a través de una VPN.

Ya sea que se use *bridges* o una VPN, a más de evitar la censura se puede ocultar el hecho de que se usa Tor. El proveedor de Internet verá que Alice se está conectando a una VPN o que está generando tráfico TLS desde la IP al *bridge*.

Es importante destacar las distribuciones GNU/Linux que vienen configuradas para trabajar con Tor. Una de las más populares es Tails que en lugar de funcionar desde el disco duro lo hace desde un DVD o una memoria USB. En caso de usar la memoria USB se puede crear una partición cifrada para conservar datos.

Tails trae el navegador Tor instalado y además provee un cliente de correo electrónico y chat configurados para que todo su tráfico viaje a través de Tor. Es una buena opción para comunicaciones secretas ya que minimiza la posibilidad de cometer errores. Tails fue la herramienta que utilizó Snowden para comunicarse con Poitras y Greenwald, los periodistas a quién entregó los documentos de la NSA.[54]

Otra distribución a tomar en cuenta es Whonix que funciona con un par de máquinas virtuales que permiten utilizar Tor de manera transparente. La primera máquina es una puerta de enlace que envía todo el tráfico a través de Tor, mientras que la segunda máquina virtual accede a la red a través la primera.

El sistema operativo Qubes ejecuta máquinas virtuales para proveer seguridad a través de compartimentación. Se puede tener una máquina virtual para el trabajo, otra para uso personal y otra para perfiles anónimos. Qubes viene con las máquinas virtuales de Whonix instaladas. Con esta

distribución en combinación con Whonix se puede acceder a una VPN a través de Tor ocultando la identidad al proveedor de VPN.[19]

Esto es útil porque muchos sitios bloquean el acceso a través de Tor o se tiene que resolver complicados *captchas* para acceder a los mismos. El autor tuvo éxito creando una cuenta de VPN mediante Bitmask[20] a través de Whonix en un sistema Qubes.

En caso de requerir seguridad en las comunicaciones, el autor recomienda utilizar Tails en lo posible ya que viene configurada para anonimato por defecto. Whonix requiere más trabajo pero es una buena opción ya que funciona como máquina virtual. Qubes tiene características de seguridad destacables pero lo recomienda para usuarios avanzados.

3.2.2 Servicios Cebollas

Los servicios ocultos de Tor, conocidos también como servicios cebollas, permiten publicar aplicaciones TCP accesibles únicamente a través de la red Tor. De esta manera se puede proveer sitios web, correo electrónico, chat, etcétera de manera anónima. Además del anonimato tiene la ventaja práctica de que se puede proporcionar servicios sin la necesidad de una dirección IP pública.

Un uso común es publicar sitios web que se pueden acceder desde el navegador Tor. Medios de comunicación de todo el mundo están utilizando este tipo de servicios para recibir denuncias de forma anónima a través de Internet. Destacan entre estos sitios el portal Wikileaks[21] o diarios como The New York Times y otros.[22]

A más de sitios web se puede utilizar los servicios cebolla para acceder a cualquier protocolo TCP que soporte SOCKS. Por ejemplo, se

19 El autor hizo las pruebas tras seguir las recomendaciones de este video: https://www.qubes-os.org/video-tours/#micah-lee-presents-qubes-os-the-operating-system-that-can-protec

20 Disponible en https://bitmask.net/

21 Buzón de denuncias disponible en: http://wlupld3ptjvsgwqw.onion

22 Sitios como The New York Times utilizan el software libre SecureDrop. Existen reconocidos medios a nivel mundial que usan este software: https://securedrop.org/directory

podría utilizar para acceder a un servidor SSH[23] de forma remota sin la necesidad de una IP pública.[55]

Para su funcionamiento el servicio oculto debe anunciar su existencia a la red. Lo primero que hace es seleccionar aleatoriamente algunos nodos y establece circuitos hasta los mismos.[24] Estos nodos serán los puntos de entrada al servicio.[56]

El servicio crea un archivo que sé llama descriptor del servicio cebolla[25] que incluye la llave pública del mismo, el *hostname* e información sobre los puntos de acceso. El *hostname* está compuesto por una secuencia de 16 caracteres del tipo abcdefghijklmnop.onion donde la parte previa a ".onion" es un *hash* de 16 caracteres derivado de la llave pública. Se firma el descriptor con la llave privada y se la sube al servidor de directorio para ser almacenado en una base de datos de *hashes*.

Si bien las direcciones ".onion" pueden ser difíciles de recordar, tienen la ventaja de que autentican la llave pública del servidor. Gracias a esto la comunicación entre un cliente y un servicio cebolla está siempre cifrada y autenticada. Una vez realizados estos pasos, el servicio está listo para esperar conexiones como se puede ver en la imagen 8.

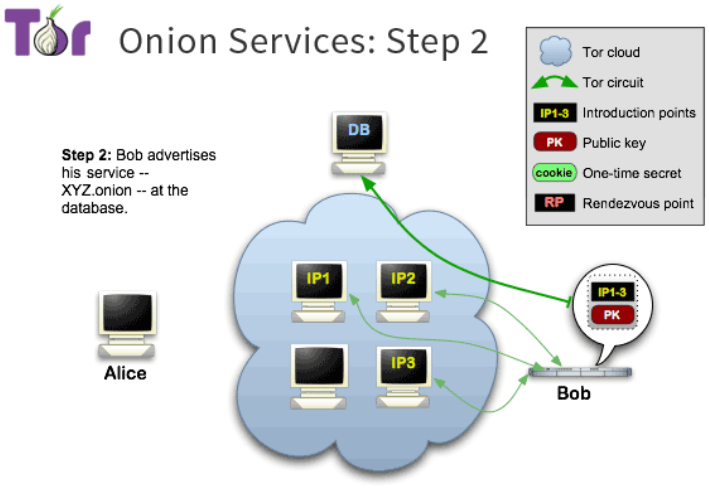

Imagen 8: Publicación del servicio oculto
Fuente: https://www.torproject.org/docs/onion-services.html.en

23 Protocolo que permite acceso remoto seguro a través de línea de comandos.
24 Un circuito es un canal atravesado por 3 nodos de la red.
25 Traducción propia del inglés: *onion service descriptor*

Si Alice quiere comunicarse con un servicio provisto por Bob entonces debe conocer la dirección del servicio cebolla. Esto puede ser porque Bob se la dio, porque encontró en un sitio web o la averiguó de alguna otra forma. Para acceder al servicio Alice hace una consulta al directorio de servicios y este devuelve el descriptor del servicio firmado por Bob. Adicionalmente Alice crea un circuito hasta un nodo conocido como punto *rendezvous*[26] como se puede ver en la imagen 9.

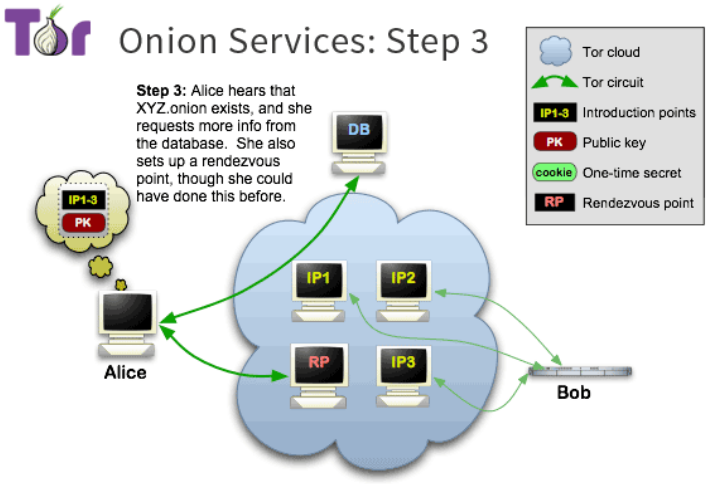

Imagen 9: Servicios ocultos.- punto de encuentro
Fuente: https://www.torproject.org/docs/onion-services.html.en

Cuando el circuito al punto de encuentro esta listo, Alice cifra un mensaje a Bob con la llave pública que incluye la dirección del punto de encuentro y la primera mitad de una negociación Diffie Hellman.

Bob recibe el mensaje con el punto de encuentro y responde al mismo a través del punto de encuentro con la segunda mitad de la negociación Diffie Hellman.

Una vez realizado estos pasos, se crea un canal al servicio oculto como se puede ver en la imagen 10. El punto de encuentro tan solo reenvía el tráfico cifrado entre Alice y Bob.

26 Los puntos de encuentro se los conoce como *rendezvous point*.

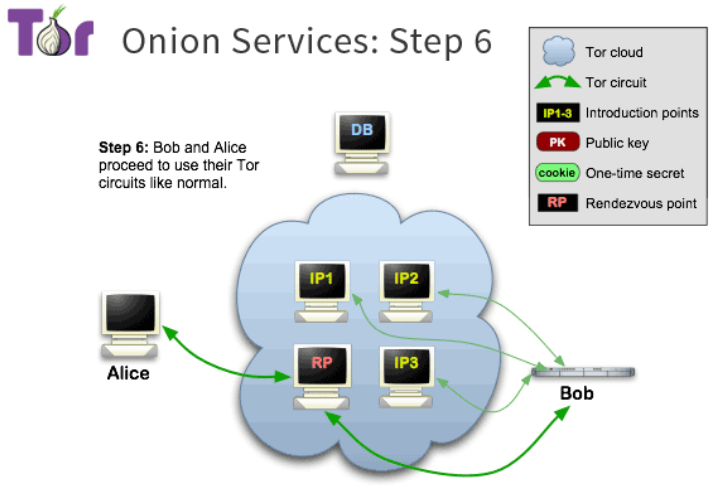

Imagen 10: Servicios Ocultos.- canal establecido
Fuente: https://www.torproject.org/docs/onion-services.html.en

3.3 El Proyecto de Internet Invisible - I2P

I2P es un proyecto creado para formar una red de comunicaciones segura y anónima. Con I2P se pueden publicar y acceder de forma anónima a servicios soportados por los protocolos TCP y UDP. Dentro de la red están disponibles varios sitios HTTP; se puede utilizar protocolos como el de Bittorrent; tener cuentas de correo electrónico anónimas; chat por IRC y más. La funcionalidad de I2P es similar a la de los servicios ocultos de Tor pero con soporte adicional a UDP. A diferencia de Tor, I2P no está pensado para anonimizar el tráfico por la red clara.[57]

Otra diferencia con Tor es que I2P no requiere servidores centralizados. Con la configuración predeterminada todos los nodos de la red funcionan como ruteadores que reenvían el tráfico a otros nodos. La selección de estos ruteadores se la accede a través de una base de datos distribuida de *hash* (DHT por sus siglas inglés).

Para la comunicación dentro de la red los miembros crean túneles de entrada y salida. Los túneles son similares a los circuitos de Tor pero de una sola dirección. Es decir existen túneles para enviar información y túneles para recibir información. Los túneles de entrada son anunciados a la base de datos descentralizada. En la imagen 11 se puede ver un ejemplo donde Alice tiene los túneles de entrada 1 y 2, mientras que Bob tiene los túneles de

entrada 3 y 4. Cuando Alice quiere mandar un mensaje a Bob escoge uno de los túneles de entrada de Bob. En el mensaje enviado Alice puede incluir el identificador de uno de sus túneles de entrada para que Bob no lo consulte en la base de datos distribuidas.

Al tener rutas diferentes el tráfico de entrada y de salida hace que sea difícil hacer un análisis del tráfico. El hecho de que todos los miembros de la red están ruteando tráfico también dificulta el criptoanálisis porque no se puede saber cuando el tráfico proviene de manera local o se está reenviando el de otro miembro de la red.

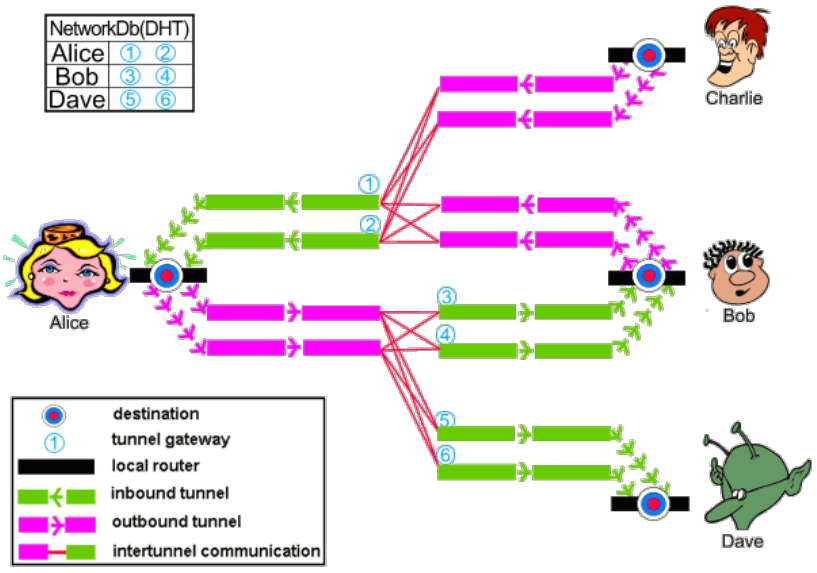

Imagen 11: I2P: Túneles
Fuente: https://geti2p.net/en/docs/how/intro

El tamaño de los túneles puede ser variable entre cero saltos hasta siete saltos. Dependiendo de esta configuración un servicio podría tener un nivel alto o bajo de anonimato.

Para las pruebas de laboratorio desarrolladas en el presente trabajo se utilizó la versión original de I2P que es desarrollada en Java. Existe además el cliente I2PD que implementa el protocolo en C++.

Una vez instalado se accede a una interfaz web que permite realizar configuraciones en el sistema y acceder a varios servicios ocultos de I2P previamente creados como se puede ver en la imagen 12.

Imagen 12: Panel de control de I2P
Fuente: Captura de pantalla

Para poder acceder a sitios web dentro de I2P es necesario configurar el *proxy* del navegador web en los puertos 4444 y 4445 de localhost para HTTP y HTTPS respectivamente. De esta manera se pueden acceder a los servicios dentro de la red oculta I2P, pero no a los sitios web públicos de Internet. Para hacer las 2 cosas a la vez es necesario utilizar sistemas como FoxyProxy,[27] Privoxy[28] o similares. También se puede utilizar I2PD Browser[29] que tiene objetivos similares a los del Navegador Tor con el cliente I2PD embebido.

3.3.1 Nombres en I2P

Para acceder a un servicio oculto se requiere la llave del mismo que está compuesta por la llave pública de 256 bytes más una llave para firmar de 128 bytes y un certificado que de momento es nulo. Esto convertido en base 64 se representa como 516 bytes. Para poder ingresar a un servicio en la red se necesita la llave en base 64 del servicio.[58]

27 Disponible en https://getfoxyproxy.org/
28 Disponible en https://www.privoxy.org/
29 Disponible en https://github.com/PurpleI2P/i2pdbrowser

De manera similar a los servicios cebolla de Tor, los servicios ocultos tienen direcciones basadas en un *hash* de la llave pública. En este caso se utiliza un *hash* SHA-256 de la llave pública en base 32 y las direcciones son del tipo: {52 caracteres}.b32.i2p. Igual que Tor el *hash* sirve para autenticar al servicio con el que se quiere acceder.

Se tiene la posibilidad de utilizar una agenda que permite administrar direcciones del tipo ".i2p" con nombres legibles similares a los dominios de Internet. Para esto hay un directorio local de nombres que mapea el nombre de dominio con el nombre completo del servicio. El funcionamiento de la agenda es similar al archivo *hosts* que existe en sistemas operativos como Linux y Windows. Adicionalmente se pueden añadir archivos hosts.txt compartidos por otros nodos de la red.

Con los nombres guardados en la agenda de direcciones se gana la posibilidad de tener nombres fáciles de recordar. A diferencia de los nombres *base32* estos no siempre significan lo mismo. En la agenda de Alice el servicio.i2p puede tener una dirección diferente del que tiene Bob en su agenda.

Los nombres agendados se pueden añadir desde otras direcciones de manera descentralizada. Añadir la agenda pública de otro directorio es un acto de confianza. De manera predeterminada I2P entrega un limitado nombres de *hosts* conocidos. El usuario tiene que añadir otros de manera manual a la agenda.

3.4 Análisis Anonimato

Las VPNs resultan útiles para proteger el contenido de las comunicaciones pero se debe depositar toda la confianza en el proveedor. El objetivo de este trabajo es que nadie sepa que Alice y Bob se están comunicando con lo cual se necesita un anonimato más fuerte que el que puede proveer este tipo de solución.

I2P tiene la característica de ser la opción más flexible para publicar servicios ocultos ya que soporta tanto TCP como UDP. Por otro lado esta no está pensada para acceder a servicios provistos en la red clara.

Tor permite crear redes obscuras propias y tener anonimato en la red clara. Esto lo hace la solución ideal para este trabajo donde se requiere anonimato en redes obscuras y la red clara. No soportar UDP podría ser un problema para las comunicaciones de voz sobre IP, sin embargo se logró solventar este inconveniente como se verá en el capítulo 6.

En los siguientes capítulos se utilizará Tor para crear seudónimos de manera anónima en la red clara y de esta manera tener cuentas de correo electrónico, chat y voz sobre IP. Para tener mayor autonomía en las comunicaciones se utilizará servicios cebollas.

4 Correo Electrónico Secreto

El correo electrónico es una de las formas de comunicación más comunes de Internet. Su desarrollo data de antes de la creación de la red de redes. Funciona de manera federada donde muchos servidores de correo tienen cuentas de usuarios y estos pueden interactuar con usuarios registrados en otros servidores de manera transparente. Es por esto que una cuenta de correo electrónico creada en el servicio de Yahoo puede comunicarse con una de Gmail o con un servidor alojado de forma autónoma por una organización.

Para el funcionamiento del correo electrónico existe una red de servidores que se comunican entre sí a través del protocolo SMTP[30]. Si Alice envía un correo desde alice@a.com a bob@b.com este correo quedará almacenado en el servidor del proveedor de correo de Bob. Bob podrá acceder a su correo electrónico a través de los protocolos POP3[31] o IMAP[32].

Si los servidores de correo electrónico no están correctamente configurados el tráfico de estas comunicaciones no es cifrado. Esto quiere decir que algún punto intermedio entre los usuarios y los servidores de correo se podría interceptar contraseñas, correos o incluso modificar los contenidos. Para proteger la comunicación entre servidores de correo y sus respectivos clientes se utilizan protocolos como TLS o STARTTLS[33]. La EFF inició el proyecto "*STARTTLS Everywhere*" con el objetivo de masificar el uso del cifrado en el transporte de correo electrónico.[59]

Este trabajo no pretende explicar como configurar correctamente un servidor de correo. En el mismo se asume que los servidores están correctamente configurados y el tráfico viaja cifrado. Incluso si los servidores están bien configurados los administradores de los mismos podrían espiar las comunicaciones. El administrador de correo de una organización

30 Protocolo de Transferencia Simple de Correo, SMTP por sus siglas en inglés.
31 Protocolo de Oficina de Correo, POP3 por sus siglas en inglés.
32 Protocolo de Acceso a Mensajes de Internet, IMAP por sus siglas en inglés.
33 Protocolo que permite añadir cifrado a protocolos inseguros sin necesidad de cambiar el puerto.

fácilmente podría leer los mensajes de los usuarios del servidor que administra.

Se quiere que Alice y Bob se puedan comunicar sin que esto implique que el administrador del servidor pueda leer sus correos o incluso sepan que los dos se están comunicando.

El cifrado extremo a extremo sirve para ocultar el contenido de las comunicaciones. La creación de seudónimos a través de Tor sirve para tener anonimato con respecto al proveedor del servicio de correo electrónico. Cuando se combina el anonimato con cifrado entre extremos, Alice y Bob pueden comunicarse sin evidenciar que lo están haciendo. El administrador del servidor sabrá que 2 personas que usan Tor se están comunicando pero no tendrá información sobre quiénes son ni el contenido de la comunicación. El proveedor local de Internet sabrá que Alice y Bob están usando Tor pero no para que.

En la sección 4.1 se explicará el funcionamiento de OpenPGP para el cifrado entre extremos; en la sección 4.2 y 4.3 el uso de seudónimos para las comunicaciones secretas; en la sección 4.4 se verá el uso de los servicios cebolla para correo electrónico; por último en la sección 4.5 se hará un análisis de comunicaciones secretas con correo electrónico.

4.1 Cifrado Extremo a Extremo con OpenPGP

Existen dos estándares que permiten proteger el contenido de los correos a través del cifrado extremo a extremo: S/MIME y OpenPGP. Los dos funcionan de forma similar al utilizar criptografía asimétrica para cifrar y firmar los mensajes. La diferencia principal es que S/MIME requiere de una autoridad certificadora y que OpenPGP utiliza una cadena de confianza para autenticar a los usuarios o en su defecto autenticación manual.

Este trabajo busca herramientas y estrategias que permitan tener comunicaciones secretas, para lo cual el anonimato es importante. Utilizar una entidad certificadora significa autenticarse ante un tercero y de esa

manera se pierda la posibilidad de ser anónimo. La cadena de confianza del protocolo OpenPGP tampoco es una opción ya que esta técnica puede exponer públicamente una red de contactos.

El estándar OpenPGP utiliza una combinación de cifrado asimétrico, simétrico y *hashing*. Existen varias implementaciones de OpenPGP, en el presente trabajo se probó GPG y OpenPGP.js. La primera es la plataforma utilizada por gran parte de los clientes de correo electrónico de escritorio mientras que la segunda es utilizada para permitir cifrado en correo *web* directamente en el navegador mediante Javascript.

El proyecto GPG tiene una guía en su sitio web donde se explica el funcionamiento del protocolo.[60] Para cifrar el mensaje Alice consigue la llave pública de Bob. Con la llave pública cifra una llave simétrica con la que se cifra el mensaje. Para descifrar el mensaje Bob utiliza su llave privada con la que descifra la llave simétrica y con esta puede leer el mensaje.

Para autenticar el mensaje, Alice se genera un *hash* del contenido del mismo y lo cifra con su llave privada. Cuando Bob recibe el mensaje genera el *hash* con el mismo algoritmo que Alice utilizó y descifra el *hash* generado por Alice con la llave pública de ella. Si los dos *hashes* coinciden entonces el mensaje no ha sido modificado y se sabe que Alice es la remitente.

Si bien el mensaje ha sido cifrado correctamente y se puede verificar con la firma de Alice ¿Cómo sabe Bob que Alice es quién dice ser? Para esto Alice debe validar la clave pública de Bob. La forma común de hacerlo es verificar manualmente la huella digital[34] de la clave que consiste de un *hash* de la llave pública compuesto por 40 caracteres. Tradicionalmente se utilizaba los últimos 8 como el identificador de la clave, sin embargo esta práctica ya no se recomienda porque se ha mostrado que es fácil generar colisiones con este identificador.[61]

Para que la autenticación funcione, la validación de las claves se debe hacer de forma presencial o buscar algún canal alternativo que sea

34 En inglés se llama *fingerprint*.

seguro para compartir. Podría ser una llamada telefónica, un mensaje de chat cifrado o publicar la huella digital completa en una cuenta de una red social donde se ha demostrado tener control; por ejemplo una cuenta de Twitter. La última opción puede poner en riesgo el anonimato si no se hace con cuidado.

Para solucionar el problema de la autenticación OpenPGP utiliza el concepto de red de confianza. En este caso Alice se encuentra con Bob en persona; con su llave privada firma la llave pública de Bob; él hace lo misma con la llave pública de ella. Podría darse el caso de Carlos que quiere contactar a Bob pero nunca lo conoció en persona; Carlos conoce a Alice quién firmó la clave de Bob y además confía en la capacidad de ella para validar las claves OpenPGP. Entonces Carlos puede descargar la clave pública de Bob, verificar la firma de Alice y de esta manera escribir a Bob con la certeza de que es la persona correcta. Si más conocidos de Carlos firmaron la clave de Bob, entonces mayor seguridad de que esa llave pública pertenece a Bob.

Esto funciona bien en organizaciones donde no se requiere el anonimato. Por ejemplo, para el proyecto Debian es un requisito para ser desarrollador tener la clave firmada por uno o más miembros de la red de confianza del proyecto de manera presencial.[62]

En el caso de proyectos como Debian tiene sentido porque se quiere saber quiénes son los desarrolladores. En el caso de comunicaciones secretas, tener la llave pública firmada por miembros de una red de confianza es una forma de exponer metadatos. Por esto el autor recomienda no firmar la clave ni subirla a los servidores de llaves públicas. Es preferible compartir la llave solamente con las personas con las que se quiere establecer comunicación y validar las claves de manera manual.[63]

Un problema importante que se debe destacar es que en el caso del correo electrónico para tener autenticación de un mensaje es mandataria la firma electrónica por lo que se tiene la característica de no repudio. Esta es una función deseable en comunicaciones públicas pero es un problema en

comunicaciones secretas porque se deja una prueba criptográfica de que Alice envió ese correo. El uso de seudónimos ayuda a mejorar en algo esta situación.

Otro problema con OpenPGP es que todos los mensajes se cifran con la misma clave, es decir que no existe secreto perfecto hacia adelante. Si un adversario consigue la clave privada de Alice y tiene sus mensajes cifrados podría leer todos los correos de ella.

En el capítulo cinco se verán protocolos como OTR, Signal y otros que proveen autenticación, repudio y secreto perfecto hacia adelante. Por lo pronto se analizará lo que se puede hacer con correo electrónico.

4.2 Correo Secreto con Seudónimos

Una estrategia para que Alice y Bob se comuniquen de forma secreta es crear cuentas anónimas en servidores de terceros a través de Tor y acceder a las mismas solo por esta red. Los mensajes enviados deben ser cifrados siempre de extremo a extremo. En la imagen 13 se puede ver como gracias a Tor alguien que espíe las comunicaciones en el servidor solo sabrá que dos cuentas anónimas se comunican entre sí a través de correos cifrados.

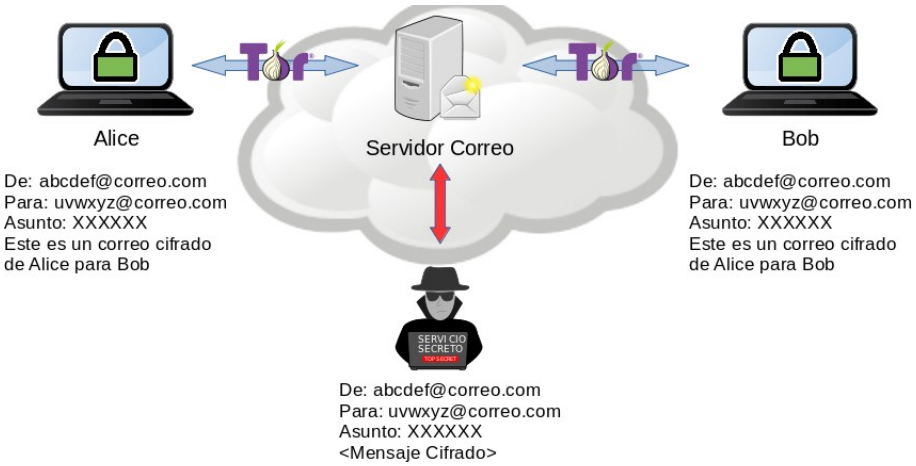

Imagen 13: Correo secreto con seudónimos
Fuente: Elaboración propia con imágenes de Openclipart y logo de Tor

Mediante la estrategia de seudónimos Alice y Bob esconden sus nombres; a través de Tor ocultan su ubicación geográfica; y mediante

OpenPGP el contenido de los mensajes. Cabe notar que generalmente el asunto de los mensajes no va cifrado[35]. Es por esto que en el correo, Alice pone un asunto sin sentido.

Al momento de crear la cuenta de correo hay que considerar que servicios como Gmail, Yahoo u Outlook, son gestionados por empresas que aparecen en el programa PRISM de la NSA. Además para crear una cuenta de correo estos servicios requiere un número telefónico. Esto es un problema para el anonimato ya que en varios países de América Latina el número de teléfono tiene que ser asociado al documento de identidad, incluso para telefonía prepaga. En el anexo uno se puede ver un listado de proveedores de correo electrónico que permiten crear cuentas de forma anónima.

Dos de estos servicios son Protonmail y Mailfence que dan la posibilidad de utilizar OpenPGP directamente desde el webmail en el navegador Tor. Esto resulta conveniente porque no se necesita utilizar software adicional. En el caso de Protonmail el código fuente de su interfaz web es libre por lo que puede ser auditado, además son de los principales mantenedores de la librería OpenPGP.js utilizada por varios proyectos de software libre.[64] [65]

Hay que tomar en cuenta que al crear una cuenta en los servidores de Protonmail no existe garantía que estén utilizando el mismo código Javascript que el que publican en sus repositorios. ¿Cómo saber que este código no fue modificado para robar la llave privada de los usuarios?

Considerando este riesgo se decidió hacer pruebas para crear una cuenta anónima con Protonmail. Este es el nivel más bajo de seguridad que se presentará en este documento pero el más sencillo. Por otro lado, que el código fuente sea libre y público es una ventaja si se quiere implementar un servidor propio. Gran parte de las características disponibles por este servicio podrían ser gestionadas de manera autónoma.

35 Desde la versión 2.0 de Enigmail es posible cifrar el asunto de los mensajes.

Protonmail asegura que todos los correos enviados entre usuarios de Protonmail están siempre cifrados entre extremos a través de OpenPGP utilizando la librería OpenPGP.js. Se puede enviar los correos cifrados a otros usuarios por fuera del servicio de Protonmail mediante OpenPGP. Por último tiene la opción de cifrar mensajes con una llave simétrica para enviar correos cifrados a gente que no usa OpenPGP.[66] Esto último resulta conveniente si se quiere enviar un correo cifrado a alguien que no tiene un cliente que soporte cifrado entre extremos.

Protonmail asegura tener “cero acceso al cifrado” de la información almacenada. Dicen que si un mensaje llega sin estar cifrado, una vez que se lo almacena en el buzón del usuario se cifra con la llave pública y se elimina el mensaje en texto plano.[67] Hay que tomar en cuenta que no hay como demostrar que esto sea así y que ellos podrían leer los correos.

Más allá de la confianza del sistema y de que Protonmail haga lo que dice hacer, toda la seguridad del sistema depende de la contraseña de la cuenta. Esta contraseña sirve para validar el usuario pero también genera un hash que se utiliza para tener acceso al buzón de correo cifrado. Es posible mejorar la seguridad activando el modo de doble contraseña, una para la cuenta de usuario y otra para el buzón.[68] Además en las pruebas de laboratorio se probó el factor de doble autenticación provisto por Protonmail que utiliza una contraseña de una sola vez. Esto es mejor que sistemas que utilizan número de teléfono porque no ponen en riesgo el anonimato.

La creación de la cuenta en Protonmail desde Tor de manera anónima resulta sencilla pero existieron inconvenientes. Para protegerse del abuso, el servicio utiliza herramientas como *captcha*, cuenta de correo de verificación, envío de SMS o donación. Para conservar el anonimato es importante no dar un número de tarjeta de crédito o telefónico para el SMS. Lo ideal es resolver el *captcha* ya que no se requiere dar ningún dato. Se podría utilizar un correo alterno siempre y cuando este se lo creo de manera anónima.

En la imagen 14 se puede ver que dependiendo del nodo por el cual se accede a Protonmail se presentará las distintas opciones de protección de abuso. En caso de no presentarse la opción de captcha o correo se puede utilizar la opción de “crear nuevo circuito Tor para este sitio” del navegador Tor hasta que obtener la opción deseada.[36]

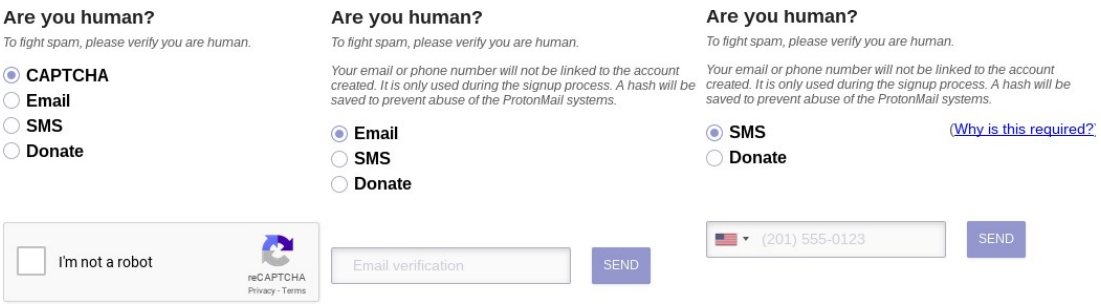

Imagen 14: Protecciones contra abuso de Protonmail.
Fuente: Elaboración propia basada en capturas de pantalla de las pruebas de laboratorio.

Es importante notar que al momento de escribir este documento es posible crear una cuenta anónima con Protonmail, esto no quiere decir que vaya a ser así siempre. Por esto es importante siempre estar pendiente de alternativas.

Protonmail tiene la posibilidad de enviar correos cifrados efímeros mediante una clave simétrica. De esta manera, por ejemplo, una fuente anónima podría enviar un correo cifrado a un periodista que no sabe proteger sus correos electrónicos. En la imagen 15 se puede ver un ejemplo de un mensaje cifrado con clave simétrica.

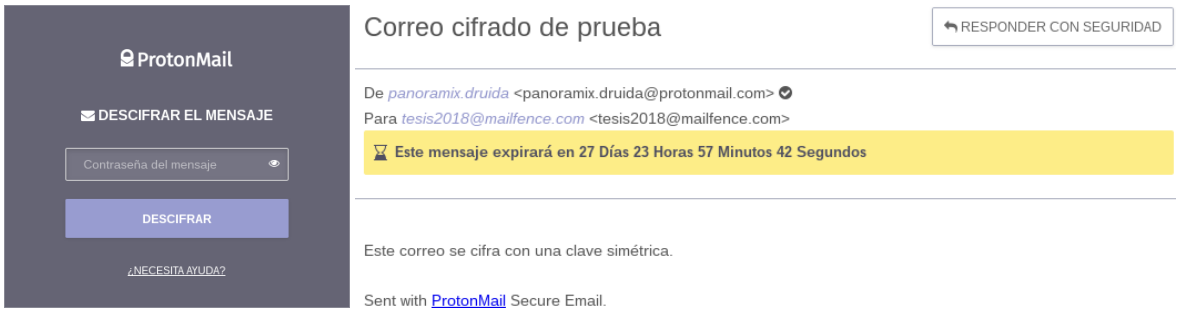

Imagen 15: Mensaje cifrado con clave simétrica con Protonmail
Fuente: Captura de pantalla

36 En el marco del “1er Encuentro Latinoamericano y Caribeño de Seguridad Informática y Privacidad en Línea” el autor explicó como crear una cuenta anónima con Protonmail: https://www.youtube.com/watch?v=CiULjdgKty0&feature=youtu.be&t=10036

Esta es una opción interesante cuando Protonmail no es un posible adversario; entonces una cuenta anónima dentro de esta plataforma puede dar un nivel aceptable de anonimato y secreto en las comunicaciones. El sitio de Protonmail no recomienda el uso de este servicio para información sensible y si el adversario es un agencia gubernamental como la NSA.[69]

4.3 Anonimato y Cifrado con Cliente Externo

Si se ejecuta un cliente de correo electrónico de forma local, la llave privada se la gestiona localmente y le será difícil al proveedor del servicio robarla. Para esto se debe buscar proveedores de correo electrónico que permitan utilizar protocolos como IMAP o POP3 seguros y sean amigables con Tor.

De manera similar a como se creó la cuenta con Protonmail, la cuenta de correo electrónico se la debe crear a través de Tor. Luego se debe buscar los datos de configuración para acceder a la misma mediante POP3 o IMAP seguro. Es preferible utilizar POP3 en lugar de IMAP ya que no crea carpetas en el servidor y es fácil de configurar para borrar los mensajes del servidor. Mientras menos información se quede en el servidor mejor.

Una vez configurada la cuenta se utilizará un cliente remoto que pueda acceder a los correos a través de Tor. Una forma fácil y segura de hacer esto es utilizar la distribución Tails. Alternativamente se hicieron pruebas con Thunderbird más los complementos Torbirdy para acceder a la red Tor y Enigmail para soporte de OpenPGP.

Una vez configurada la cuenta de correo en el cliente externo solo se accederá a la interfaz web del proveedor del correo para tareas como la actualización de contraseña[37]. La administración del correo electrónico se realizará siempre desde el cliente. De esta manera la única información que

37 También se puede ingresar para verificar que los correos están cifrados puesto que los mismos deberían ser ilegibles desde la interfaz web.

tiene el proveedor del correo electrónico son los metadatos de las comunicaciones siguiendo el esquema de la imagen 13.

4.4 Correo Electrónico y Servicios Ocultos

Con el esquema anterior se tiene anonimato y secreto en las comunicaciones pero no se tiene autonomía. Un servidor de correo electrónico propio brinda autonomía, sin embargo el nivel de anonimato no será bueno ya que el dominio estará asociado a la tarjeta de crédito de quién lo compró y se necesita una dirección IP pública.

Para tener anonimato en las comunicaciones y autonomía a la vez se puede configurar un servidor de correo electrónico como un servicio oculto de Tor o I2P. De esta manera Alice y Bob pueden comunicarse de manera segura sin revelar su ubicación, con cifrado extremo a extremo y utilizando infraestructura propia.

Ejecutar un servidor propio y publicarlo como servicio cebolla se lo puede hacer en hardware de pocos recursos como un Raspberry PI o una computadora vieja. Además al ser un servicio oculto no es necesario tener una dirección IP pública ni usar DNS. El servicio oculto puede ser autónomo para un grupo de gente que requiere comunicarse entre si de forma segura, como es el caso de una organización

En las pruebas de laboratorio se logró federar los servidores de correo electrónicos a través de Tor sin necesidad de usar servidores de DNS. De esta manera Alice puede tener su servidor y Bob el suyo como se puede ver en la imagen 16. Claro que si los dos servidores no están conectados a la vez se tendrá problemas con la entrega de correos.[38]

No se explicará como configurar un servidor de correo electrónico ya que esta fuera del alcance de este documento y existe mucha documentación en línea al respecto. Lo que se explicará es la configuración básica de servicios cebolla.

38 Se hicieron pruebas siguiendo las instrucciones mostradas en http://l6eotn7hkrq6a4n3.onion/2016/mail-local-por-tor/ (Accesible solo por Tor)

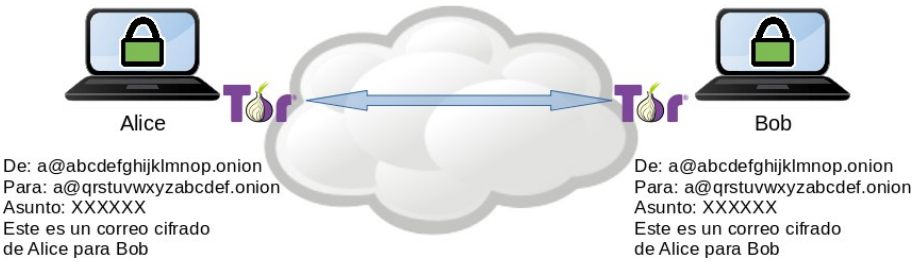

Imagen 16: Correo federado con servicios cebolla
Fuente: Elaboración propia con gráficos de Openclipart y logo de Tor

La primera opción es crear un servicio oculto para acceder a través de *webmail* que soporte cifrado entre extremos. En este laboratorio se utilizó la interfaz de Roundcube pero lo ideal sería utilizar algo como Protonmail ya que brinda cifrado extremo a extremo de manera transparente. La configuración más sencilla de Tor en el archivo /etc/tor/torrc sería algo como:

```
HiddenServiceDir /var/lib/tor/correo
HiddenServicePort 80 127.0.0.1:80
HiddenServicePort 25 127.0.0.1:25
```

La primera línea dice donde se guardará las llaves criptográficas del servicio oculto; la segunda dice que el puerto 80 local (127.0.0.1) será publicado como servicio cebolla para el acceso por *webmail*. La tercera línea es útil si este servidor se conectará a otros servidores de correo.

En la carpeta /var/lib/tor/correo se crearán dos archivos. El primero llamado *private_key* que contiene el par de llaves públicas y privadas del servicio. El segundo se llama *hostname* que tiene la dirección ".onion" del servicio oculto.

Si el servidor de correo electrónico y el servidor web están configurados para escuchar solamente en 127.0.0.1, entonces en la red local no se podrá saber que se está ejecutando un servidor de correo electrónico. Se podrá saber que una máquina está generando tráfico dentro de la red Tor pero no para que.

En la imagen 17 se puede ver el resultado de la configuración cuando se accede a la interfaz de *webmail*. Si bien está publicada en el puerto 80 la comunicación es cifrada a través del servicio oculto de Tor.

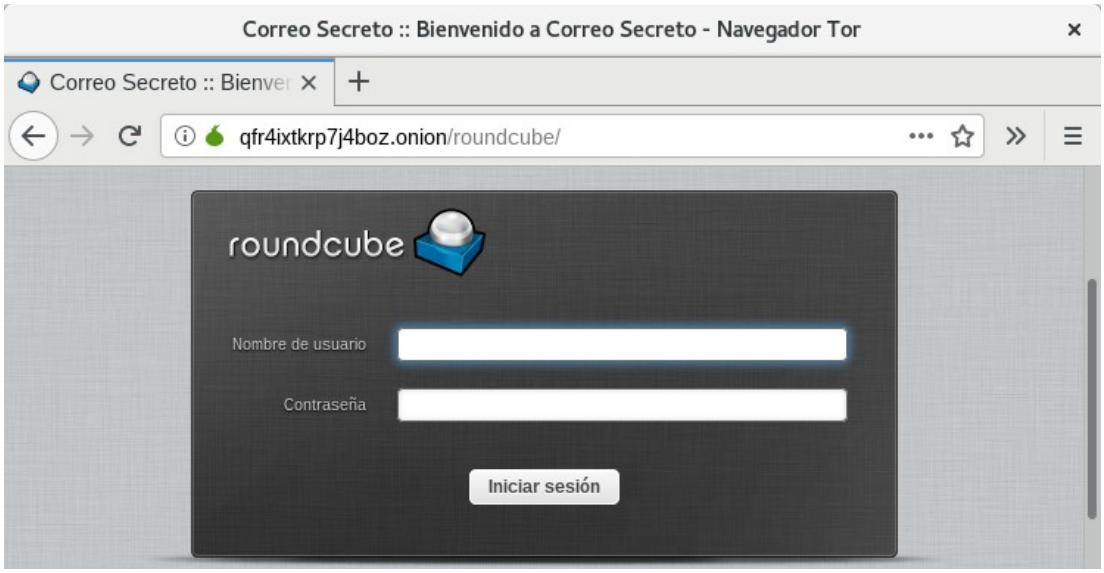

Imagen 17: Acceso a *webmail* mediante servicio oculto de Tor
Fuente: Captura de pantalla de las pruebas de laboratorio.

En este caso se utiliza el mismo servicio oculto para la interfaz web y correo electrónico. Se puede configurar dos servicios ocultos diferentes, una para el correo electrónico y otro para el servidor web cada uno con su propia dirección ".onion". De esta manera la dirección de correo electrónico no revela ninguna información sobre como acceder al correo web. La configuración sería la siguiente:

```
HiddenServiceDir /var/lib/tor/correo
HiddenServicePort 25 127.0.0.1:25

HiddenServiceDir /var/lib/tor/web
HiddenServicePort 80 127.0.0.1:80
```

Se pueden utilizar los servicios ocultos para acceder a las cuentas de correo electrónico a través de POP3 o IMAP. En este caso la configuración sería la siguiente:

```
HiddenServiceDir /var/lib/tor/correo
HiddenServicePort 25 127.0.0.1:25

HiddenServiceDir /var/lib/tor/envioRecepcion
HiddenServicePort 110 127.0.0.1:110

HiddenServicePort 587 127.0.0.1:587
```

Un cliente de correo electrónico con la capacidad de acceder a la red Tor se lo puede configurar con el *hostname* del segundo servicio oculto para enviar correos a través del puerto 587 y recibir correos a través del puerto 110. En este caso no se está utilizando STARTTLS o TLS porque el servicio oculto cifra y autentica el canal.

4.5 Análisis Correo Electrónico

Existen varias formas en las que se puede tener comunicaciones secretas a través del correo electrónico. La primera opción que se vio es el uso de servicios como Protonmail que bien utilizados pueden dar una buena expectativa de anonimato pero requieren tener confianza total en el proveedor del servicio.

La segunda opción fue utilizar proveedores de servicios que permitan crear cuentas accesibles a través de POP3 o IMAP seguro. En este caso se debe acceder siempre a la cuenta a través de Tor y cifrar siempre los mensajes.

La última opción es configurar un servidor propio de correo electrónico y publicarlo como servicio oculto. En este caso se gana autonomía pero se debe confiar en el administrador del servidor de correo electrónico ya que quién administre el mismo podría saber a quién pertenecen las cuentas. En el caso de organizaciones en lugar de confiar en un desconocido deben depositar la confianza en alguien que pertenece a la misma organización o que al menos es conocido de la misma.

Es importante notar que si bien el uso de seudónimos permite tener cierto nivel de repudio en el envío de mensajes firmados, estos están criptográficamente asociadas el seudónimo de quien envía el mensaje. Si se logra demostrar que Alice es dueña de la cuentaabcdef@correo.com entonces todos los mensajes firmados serán asociados a Alice.

Otra debilidad del cifrado para mensajes de correo electrónico es que todos los mensajes se cifran con la misma llave. Es por esto que se

recomienda rotar la llave periódicamente, lo cual no siempre es una tarea sencilla ya que se debe volver a distribuir la llave pública.

5 Chat Secreto

La mensajería instantánea, también conocida como chat, permite comunicarse de una forma rápida a través de mensajes cortos de texto. Uno de los primeros protocolos que todavía se utiliza es IRC; este fue creado en 1988 y llegó a ser un estándar RFC en el año 1993. Desde sus inicios el protocolo fue federado donde varios servidores forman redes a las que se pueden conectar las personas. [70] [71]

Con el tiempo esto cambió y sistemas de chat como ICQ, MSN Messenger, chat de Yahoo y otros se convirtieron en sistemas centralizados. Un usuario de MSN Messenger no podría chatear con uno de Yahoo. En la actualidad la comunicación por chat se ha vuelto popular a través de los teléfonos móviles mediante servicios centralizados como Whatsapp o Telegram.

En 1998 Jeremie Miller desarrolló el sistema Jabber, hoy conocido como XMPP, para ser una alternativa descentralizada a los sistemas utilizados en la época. En 1999 se empezó a trabajar para que Jabber se convirtiera en un estándar IETF. En 2004 se publicaron los RFC 3920 y 3921 de la especificación, los mismos que fueron revisados en 2011 y constan como los RFC 6120, 6121 y 6122. En la actualidad existen millones de personas que se comunican a través de XMPP.[72]

A pesar de que XMPP es un protocolo estándar de la IETF y que tiene varias implementaciones libres para clientes y servidores no es tan utilizado como plataformas centralizadas. Para enero de 2018 eran más de mil quinientos millones las personas con cuenta registrada en Whatsapp.[73] Según Pavel Durov, fundador de Telegram, en marzo de 2018 esta plataforma tenía doscientos millones de usuarios activos.[74]

Para registrar un usuario en los servicios de Whatsapp o Telegram es necesario utilizar un número celular. Esto es un problema para el anonimato ya que en países como Argentina,[75] Ecuador[76] y otros el número celular se asocia al documento de identidad.

Empresas proveedoras de servicios como Whatsapp o Telegram pueden recolectar los metadatos de cientos de millones de personas, incluso sin conocer el contenido de los mensajes. En los servicios federados son los administradores de cada servidor los que pueden recolectar el contenido de las comunicaciones y sus metadatos. Mientras existan menos usuarios por servidor, menor poder tendrán los proveedores de los servicios.

A diferencia del correo electrónico, las comunicaciones por chat no están estandarizadas por lo que existen varias soluciones para comunicaciones secretas. El objetivo de este trabajo no es analizar todas las soluciones, sino demostrar que es posible chatear de manera secreta en Internet.

Por este motivo se priorizó el protocolo de chat XMPP en combinación con Tor para el anonimato y OTR para cifrado extremo a extremo. En la sección 5.1 se explicará brevemente el funcionamiento de XMPP y la forma en la que se puede crear una cuenta anónima. Más adelante en la misma sección se verá como implementar un servicio oculto de XMPP para tener autonomía.

En la sección 5.2 se verá el protocolo OTR para la protección del contenido de las comunicaciones; en la sección 5.3 el protocolo Signal y su adaptación para funcionar con XMPP.

En la sección 5.4 y 5.5 se revisarán los protocolos Ricochet y Briar que funcionan de manera descentralizada por lo que permiten la comunicación entre pares (P2P por sus siglas en inglés) sin la necesidad de servidores. Así se elimina el riesgo de tener un administrador de servicio como adversario.

En la sección 5.6 se verá Delta Chat que utiliza OpenPGP y los protocolos de correo electrónico como una forma de mensajería instantánea.

5.1 Anonimato con XMPP

En 1999 se creó el protocolo XMPP[39] que fue aprobado como norma IETF en el año 2004. El protocolo tiene características similares a la del correo electrónico; una cuenta XMPP es del tipo de usuario@dominio.com y se utilizan registros especiales de DNS para conectar los servidores entre sí.

En la imagen 18 se puede ver el esquema de federación de XMPP. Nótese que en el mismo se encuentra Google Talk, el sistema de chats que utilizó Google hasta 2013 cuando lo cambió por Hangouts.[77] Hasta ese entonces las cuentas de correo de Gmail funcionaban también como cuentas de chat XMPP.

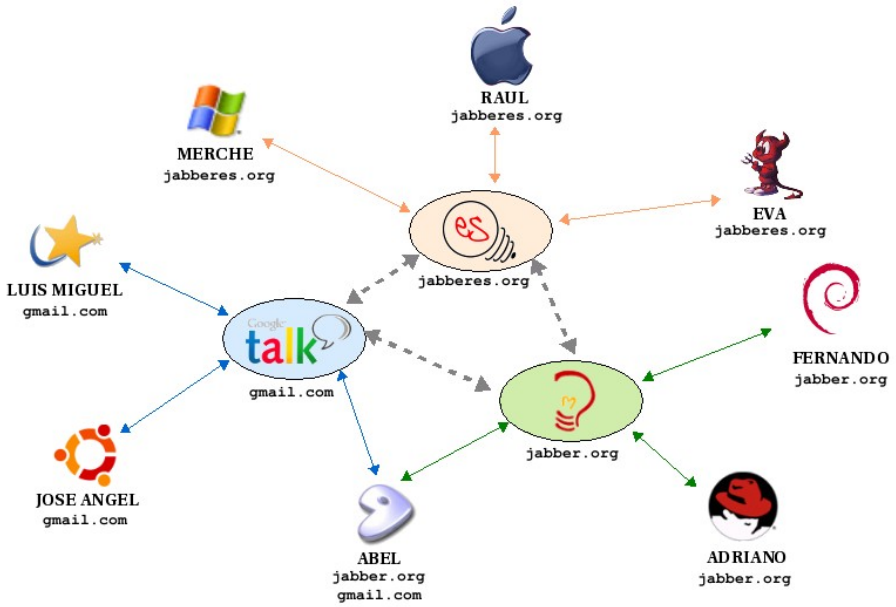

Imagen 18: Federación en XMPP
Fuente: https://www.jabberes.org/introduccion

A pesar de que empresas como Google ya no soporten este protocolo; existen cientos de servidores que sí lo hacen y en muchos de ellos se pueden crear cuentas de forma anónima.

39 En ese entonces conocido como Jabber.

5.1.1 Cuentas Anónimas de chat con XMPP

Se utilizará la estrategia de seudónimos a través de Tor para tener anonimato con respecto al proveedor del servicio. La lógica es similar a la de correo electrónico descrita en la imagen 13 con la diferencia de que los mensajes de chat no contienen asunto.

Dependiendo del proveedor y el cliente de chat se puede crear la cuenta a través del navegador Tor o desde la aplicación. En el anexo 3 se puede ver un listado de aplicaciones compatibles con XMPP y el soporte de cifrado entre extremos.

Para crear la cuenta de forma anónima se recomienda utilizar la distribución Tails. Como se mencionó en el capítulo tres, esta distribución envía todo su tráfico a través de la red Tor. Adicionalmente viene con el cliente de chat Pidgin que soporta varios protocolos de chat, entre los que se incluye XMPP. Tails trae este cliente configurado para funcionar con el protocolo de cifrado entre extremos OTR que se describirá en la sección 5.2

Alternativamente a Tails, una buena opción para utilizar XMPP con anonimato es el cliente de escritorio multiplataforma CoyIM que esta disponible para Windows, Linux y Mac. Este aplicativo tiene la característica de enviar su tráfico por la red Tor y utilizar OTR de manera predeterminada. CoyIM permite registrar cuentas de forma anónima en por lo menos cuatro proveedores como se puede ver en la imagen 19.

Además de los servidores provistos por aplicaciones como CoyIM, existen listados en la web[40] de otros servidores públicos donde se puede crear cuentas. Tails y CoyIM son dos opciones con las que se puede crear cuentas anónimas, pero no las únicas.

40 Un listado extenso se puede encontrar en https://list.jabber.at/

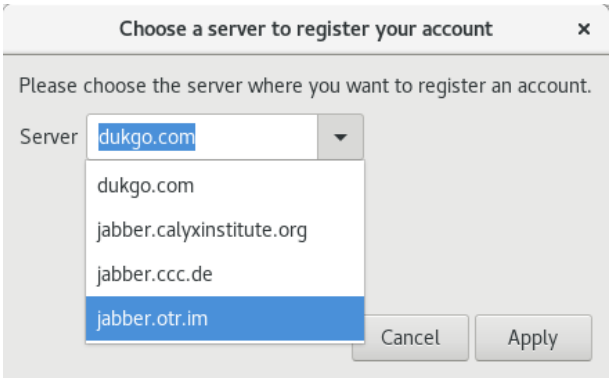

Imagen 19: Crear cuenta de chat desde CoyIM
Fuente: Captura de pantalla

5.1.2 XMPP como Servicio Cebolla

XMPP funciona a través de TCP por lo que puede funcionar como un servicio cebolla. En la materia de Redes I, el autor en colaboración de dos compañeros maestrantes implementaron un sistema de chat XMPP publicado como servicio cebolla.

Si bien se requiere conocimientos técnicos en Linux, una implementación básica del sistema resultó simple. Se instaló el servidor libre Ejabberd y se lo configuró para que escuchara exclusivamente en la dirección IP 127.0.0.1. Es decir que el servicio es accesible solamente de manera local por lo que no será visible en la red LAN.

La lógica para configurar el servicio oculto es la misma que la de correo electrónico donde se define los puertos a publicar. XMPP utiliza el puerto 5222 para comunicarse con los clientes. Ejabberd utiliza el puerto 5280 para su configuración. Además resulta conveniente poder acceder al servidor a través del protocolo SSH para la administración. En el archivo de configuración /etc/tor/torrc se añadieron las siguientes líneas.

```
#Servicio oculto de chat
      HiddenServiceDir /var/lib/tor/xmpp/
      HiddenServicePort 5222 127.0.0.1:5222

#Servicio oculto para administración
      HiddenServiceDir /var/lib/tor/xmpp-admin/
```

```
HiddenServicePort 5280 127.0.0.1:5280
HiddenServicePort 22 127.0.0.1:22
```

El primer servicio oculto sirve para ingresar con las cuentas de chat desde clientes como Pidgin o CoyIM a través de la dirección “.onion”; la misma deberá ser distribuida entre todos los usuarios. El segundo servicio oculto será accesible solamente por los administradores del servicio. En la imagen 20 se puede ver la interfaz de configuración de Ejabberd desde el navegador Tor.

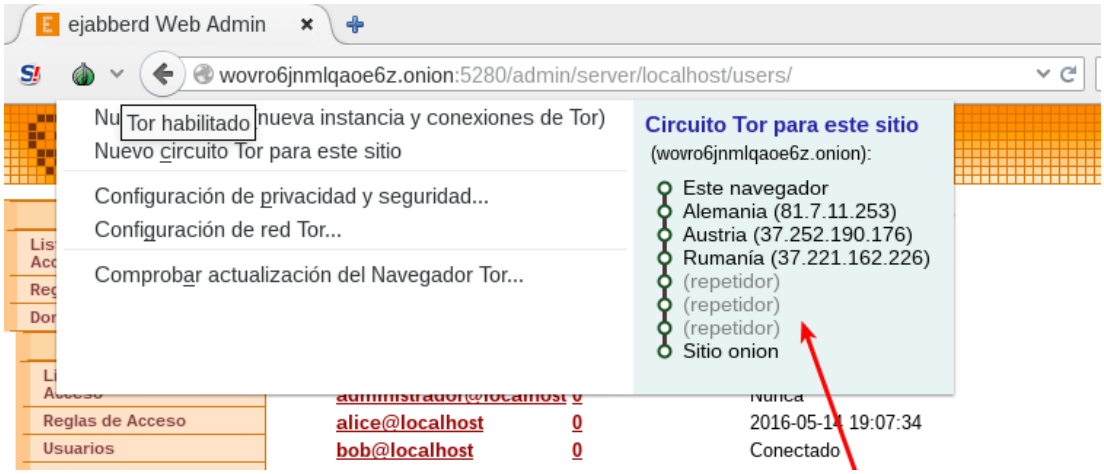

Imagen 20: Interfaz administrativa de Ejabberd a través de dirección “.onion”
Fuente: Captura de pantalla realizada en el proyecto final de Redes I

Para acceder al servicio oculto, basta con utilizar un cliente XMPP que se conecte a la red Tor. Adicionalmente a Pidgin y CoyIM, existen clientes para Android como Conversations o Pix-Art Messenger a través de Orbot. En la imagen 21 se puede ver la configuración del cliente para Android Conversations.

Tanto la imagen 21 como la 20 corresponden a una misma prueba de laboratorio. Como se puede observar el dominio para acceder a la interfaz de administración es distinto al utilizado para ingresar con la cuenta.

Tener un servicio oculto de Tor para XMPP es una buena opción para organizaciones que quieren tener independencia en sus comunicaciones. ¿Pero qué pasa si Alice trabaja en la organización X y Bob trabaja en la organización Y? ¿Es posible federar el protocolo XMPP a través de servicios ocultos?

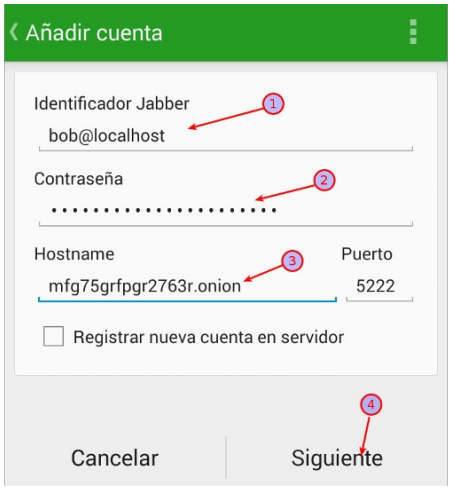

Imagen 21: Configuración Conversations y servicio cebolla
Fuente: Elaboración propia por captura de pantalla

Sobre este punto no se realizaron pruebas pero es importante destacar que existe software que dice soportar esta funcionalidad. Este es el caso del módulo mod_onions disponible para el servidor XMPP libre Prosody.[78] De esta manera alice@abcdefghijklmnpq.onion podría comunicarse con bob@zxywvutsrqomnlkj.onion de forma transparente con un esquema similar al que se puede ver en la imagen 16 del capítulo 3.

Tanto Alice como Bob deben confiar en el administrador del servidor, salvo que Alice y Bob gestionen sus propios servicios ocultos. Thijs Alkemade, desarrollador del cliente de chat Adium, en 2013 propuso la idea de que clientes de chat como Adium o Pidgin tengan complementos que permitan embeber un servidor XMPP federado con servicios ocultos de Tor. [79]

Si se llegara a implementar esta solución ya no existiría un administrador de servidor de quién defenderse. Adicionalmente tanto Alice como Bob deben estar conectados de manera simultánea para poder chatear. Más adelante en es te capítulo se verán soluciones como Briar y Ricochet que permiten comunicarse sin la necesidad de servidores.

XMPP no cifra los mensajes entre extremos de manera predeterminada. Hasta ahora se ha visto como crear cuentas anónimas, en la próxima sección se verá el protocolo OTR para el secreto en el contenido de los mensajes.

5.2 Cifrado Extremo a Extremo con OTR

El protocolo OTR permite añadir cifrado entre extremos a sistemas que no tienen esta funcionalidad; cómo es el caso de XMPP pero podría ser otro protocolo. A diferencia de OpenPGP, OTR tiene la propiedad de secreto perfecto hacia adelante. Es decir cada mensaje se cifra con una clave diferente. Para lograr esto en lugar de cifrar los mensajes con la llave pública se negocia una llave simétrica efímera para cada mensaje a través de Diffie Hellman.[38] De esta manera, si un adversario logra acceder a la llave de cifrado del mensaje, solo podrá leer este mensaje y ninguno más.

El protocolo se basa en establecer el cifrado de los mensajes desde la primera interacción; así estos no estén autenticados. Una vez que la sesión de chat se inició, se utilizan llaves públicas para la autenticación de las partes de la conversación. Cada mensaje, a más del contenido de la comunicación, incluye la negociación Diffie Hellman para el próximo mensaje.[80]

Utilizar criptografía asimétrica para autenticar los extremos de la conversación y criptografía simétrica para cifrar los mensajes permite tener comunicaciones autenticadas y repudio de manera simultánea. En este caso Alice puede conversar con Bob y si verificaron sus llaves públicas tienen la certeza de que se están comunicando entre sí. Alguien que logre interceptar las comunicaciones no podrá probar criptográficamente que el mensaje enviado por Alice fue enviado por ella, ni siquiera Bob lo puede hacer. Esta es una diferencia importante si se compara con OpenPGP donde un adversario o Bob tendrán mensajes firmados criptográficamente por Alice.

Cuando se inicia una conversación cifrada en una aplicación cliente, se muestra un mensaje de que la sesión de chat es cifrada pero no autenticada como se puede ver en la imagen 22. A la izquierda se muestra un ejemplo con el cliente de chat para Android Conversations y a la derecha el cliente Tor Messenger[41].

41 Las capturas fueran realizadas en 2016, Conversations ya no soporta OTR y Tor Messenger ya no se desarrolla. Estas capturas sirven para ilustrar el funcionamiento del protocolo que funciona parecido en aplicaciones como Pidgin, Adium, CoyIM y otras.

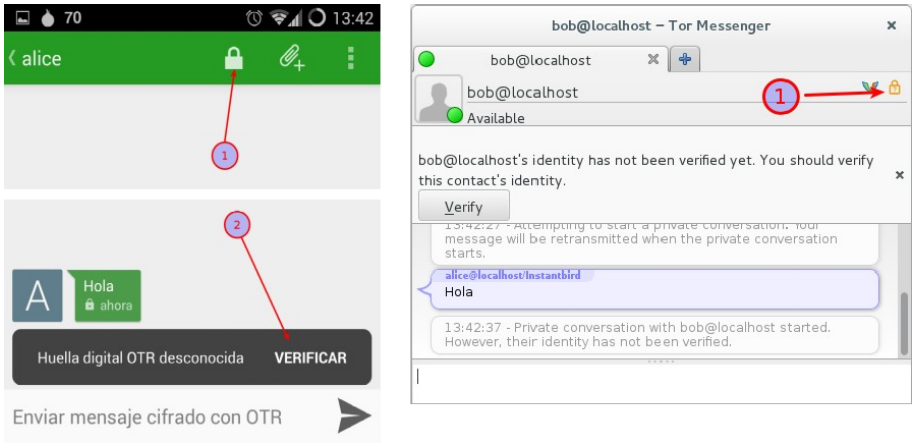

Imagen 22: Verificación de mensajes de OTR
Fuente: Elaboración propia a través de capturas de pantalla.

Una vez que se inició la conversación cifrada la forma común de autenticar es verificando la huella digital de la llave pública que consiste de un *hash*. Adicionalmente aplicaciones como Pidgin permiten abstraer esta verificación utilizando pregunta y respuesta o secreto compartido. De esta manera Alice pregunta a Bob algo que solo los dos conocen y si la respuesta es correcta Alice sabe que está hablando con Bob. Bob deberá hacer lo mismo para autenticar a Alice. El secreto compartido consiste en una frase que tanto Alice y Bob se pusieron de acuerdo previamente. Esta verificación se debe hacer solamente la primera vez que Alice conversa con Bob.

En la imagen 23 se puede ver captura de pantalla de Tor Messenger donde se muestra a la izquierda la validación por huella digital y a la derecha la opción de pregunta y respuesta.

Las características de repudio como secreto perfecto pueden ver vulneradas cuando se conservan los mensajes de manera local. Si bien los mismos no están firmados como sucedería en el correo electrónico, sí están almacenadas y podrían ser leídos por un tercero con acceso físico al dispositivo. Este problema se agrava cuando a más de registrar conversaciones se las respalda en la nube.

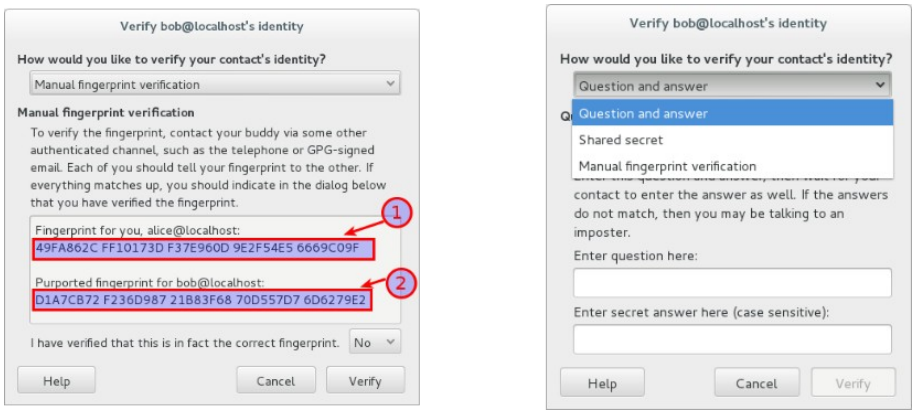

Imagen 23: Autenticación OTR
Fuente: Elaboración propia a través de captura de pantalla

Este es el caso de la aplicación Whatsapp que de manera predeterminada esta configurada para respaldar periódicamente las conversaciones en los servidores de la empresa. [81] Es probable que Alice borre sus mensajes y no los respalde en la nube, esto no quiere decir que sus contactos hagan lo mismo. Cabe recordar que Whatsapp pertenece a Facebook y este último es parte del programa PRISM de la NSA.

Al momento de escribir este documento se encuentra avanzado el desarrollo de la versión 4 de OTR. Sofía Celi, una de las desarrolladoras de la versión 4 de el estándar, explica en la lista de correo de desarrolladores de OTR que en esta versión se añade la opción de mensajes asincrónicos, se actualizan las primitivas criptográficas y otras mejoras.[82]

Más allá de la parte técnica, es importante destacar que la versión 4 de OTR y CoyIM son proyectos desarrollados con gente de América Latina. El desarrollo es liderado por la organización Centro de Autonomía Digital, con sede en Ecuador y España, que trabaja “con el propósito de hacer de Internet un lugar más seguro para todas las personas.”[83]

5.3 Signal y Derivados

Signal es una aplicación de chat seguro con cifrado entre extremos para teléfonos celulares que también se puede utilizar en computadoras. Según la página web de la aplicación, personajes como Edward Snowden,

Laura Poitras, Bruce Schneider y Matt Green recomiendan utilizar Signal para comunicaciones seguras.[84]

Signal es una aplicación cliente servidor que adapta el protocolo OTR y entre otras mejoras permite iniciar sesiones de chat de manera asincrónica. Este es un beneficio importante para la usabilidad de chat a través de dispositivos móviles. Además el protocolo permite tener chats grupales con cifrado entre extremos.

Es una aplicación buena para ocultar el contenido de los mensajes pero mala para proteger el anonimato. En primer lugar es un servicio centralizado en el cual el proveedor tiene la capacidad de monitorear y registrar los metadatos de la comunicación.

En segundo lugar Signal utiliza el número de teléfono para autenticar la cuenta de los usuarios. En países como Ecuador[76] y Argentina[75] es necesario asociar el número telefónico con el documento de identidad. Esto es un problema para el anonimato porque los administradores del servicio Signal podrían estar espiando el tráfico.

Incluso si los administradores del servicio de Signal tienen un fuerte compromiso para proteger el anonimato, este se pone en riesgos en los chats grupales. Si Alice vive en un país totalitario y utiliza los grupos de Signal para organizarse entonces cualquier persona dentro del grupo va a conocer los números de teléfonos de todos los demás miembros. Si hay un infiltrado dentro del mismo, este podrá saber quién es el dueño de cada número de teléfono y llevar un registro de qué es lo que dicen en el grupo. De poco sirve el secreto futuro perfecto en estas situaciones. En la imagen 24 se puede ver el ejemplo de un mensaje enviado a un grupo de Signal donde se muestra el número telefónico de quién envía el mensaje.

Imagen 24: Mensaje enviado a un chat grupal de Signal
Fuente: Elaboración propia.

Existen trucos para evitar utilizar el número de teléfono de la operadora local mediante un número de teléfono desechable o utilizar un número telefónico de voz sobre IP como lo explica Micah Lee en su artículo para The Intercept.[85]

No es objetivo de este trabajo describir como crear una cuenta anónima con Signal pero vale la pena destacar que es posible utilizar un número telefónico distinto al de la operadora para activar una cuenta; esto aplica también para sistemas como Telegram o Whatsapp.

5.3.1 Protocolo Signal

La baja expectativa de anonimato que provee Signal no la hace recomendable para este trabajo; sin embargo vale la pena analizar su protocolo. El mismo es utilizado por soluciones propietarias como Whatsapp, Skype, Google o Facebook;[86] pero también se lo ha adaptado en soluciones libres como OMEMO, Matrix y Wired.

Moxie Marlinspike, creador de Signal, explicaba en el año 2013 que protocolos como OTR proveen la propiedad de secreto perfecto hacia adelante y que funcionan bien en mensajería sincrónica. La mensajería en aplicaciones celulares funciona de manera asincrónica más que sincrónica. [87] Es por esto que uno de los aportes importantes es el poder iniciar una sesión de chat sin la necesidad de que las dos aplicaciones estén conectadas a la vez.

Marlinspike sostenía en 2013 que para tener chat seguro en dispositivos móviles se debía escoger entre secreto perfecto adelante o conversaciones asincrónicas. Citaba como ejemplo a aplicaciones como Threema que usan un esquema similar al de OpenPGP donde todas los mensajes se cifran con la misma llave; de esta manera es fácil tener comunicaciones asincrónicas pero se sacrifica el secreto perfecto hacia adelante.

Para solucionar este problema, un usuario de Signal crea su par de llaves públicas y privadas con las que firma 100 prenegociaciones Diffie

Hellman que son subidas a un servidor. Si Alice quiere contactar a Bob descarga una prenegociación Diffie Hellman firmada con la llave pública de Bob, envía su parte de la negociación firmada y el mensaje cifrado sin la necesidad de que Bob este conectado. Es importante verificar las llaves públicas ya que las firmas en las prenegociaciones sirven para protegerse de ataques de hombre en el medio.

Las claves de prenegociación se utilizan una sola vez y se actualizan periódicamente. Para el funcionamiento correcto del protocolo Signal es necesario tener un servidor de confianza que almacene las llaves públicas de los usuarios así como sus lotes de prenegociación Diffie Hellman.

Otro aporte importante del protocolo Signal es la capacidad de tener grupos de chat con comunicación cifrada extremo a extremo. Además la información de los grupos no se guarda en el servidor sino que es distribuida entre los miembros de los mismos.[88]

El código fuente de la aplicación Signal es libre y en el caso de Android permiten verificar que el mismo corresponde al binario que se distribuye a través de Google Play o al que se puede descargar desde su página web.[89] Esto es bueno porque se puede comprobar que el software hace lo que dice hacer.

El código fuente del servidor también es libre y se lo puede descargar desde GitHub.[90] A diferencia del cliente, no se puede verificar que la implementación de los servidores de Signal corresponden al código fuente publicado.

Los protocolos utilizados para el funcionamiento de Signal están documentos y además existen librerías en Java, C y Javascript que permiten adaptar el mismo a otros sistemas de chat. Este es el caso de OMEMO para XMPP, OLM para Matrix y para aplicaciones como Wired.

Se invita al lector a investigar sobre estas herramientas, en el caso de este trabajo se hablará sobre OMEMO ya que permite añadir los beneficios vistos en Signal a un protocolo federado como XMPP.

5.3.2 OMEMO: Multi-End Message and Object Encryption

OMEMO es una adaptación del protocolo Signal para ser utilizado en el estándar XMPP. Su funcionamiento es parecido al de los grupos de Signal. Según una auditoría realizada al protocolo, cada dispositivo es un participante del grupo.[91] De esta manera Alice podría tener un teléfono y una computadora conectadas a una sesión de OMEMO, mientras que Bob está conectado desde un celular. Alice puede seguir la conversación de forma transparente desde cualquier de los dispositivos.

En la imagen 24 se puede ver un gráfico en el que Alice tiene conectado una computadora y un celular a un servidor XMPP, mientras que Bob tiene conectado solo el celular. Si Bob quisiera añadir su computador a la sesión debería subir un identificador al servidor en conjunto con la llave pública y las prenegociaciones de Diffie Hellman.

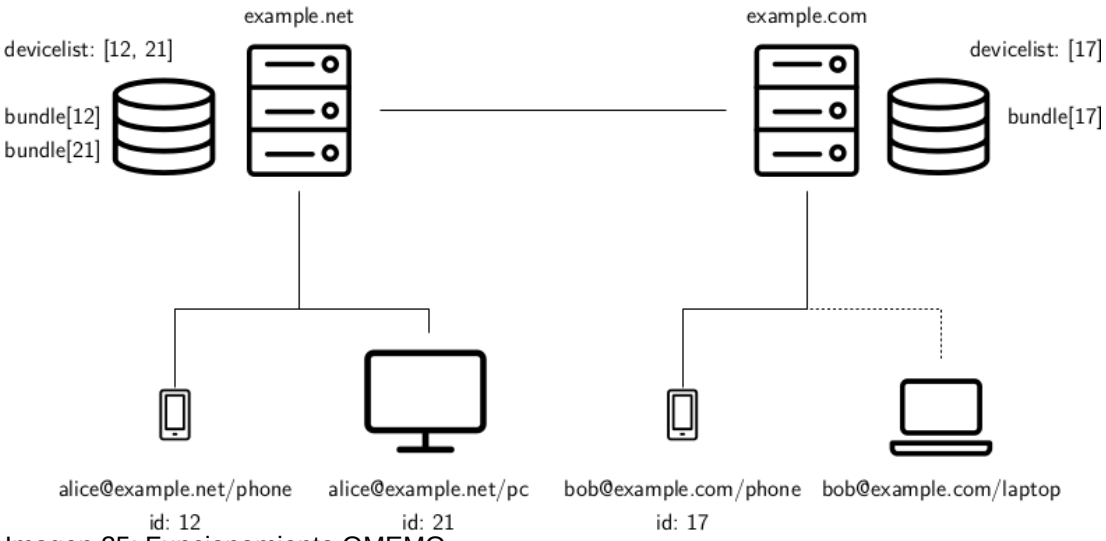

Imagen 25: Funcionamiento OMEMO
Fuente: https://conversations.im/omemo/audit.pdf

La capacidad de tener varios dispositivos conectados a una solo cuenta es positivo desde el punto de vista de movilidad y usabilidad. Alice podría iniciar una conversación en su computadora y cuando sale de casa la puede continuar desde su celular. La auditoría sostiene que esto tiene implicaciones de seguridad. El hecho de que varios dispositivos puedan compartir el historial de una conversación trae el riesgo de que si un

dispositivo este comprometido se compromete la seguridad de toda las conversaciones del usuario.

Hay que tomar en cuenta que al utilizar XMPP los administradores de los servidores podrían saber los metadatos de las comunicaciones de sus usuarios. Esto se puede resolver creando cuentas anónimas a través de Tor en servidores públicos o implementando servidores propios de chat XMPP como se vio en la sección 5.1.

A diferencia de OTR, OMEMO tiene que estar soportado por el servidor y para esto se deben buscar servidores que tengan esta funcionalidad[42]. En caso de implementar un servidor XMPP propio, el mismo debe soportar las extensiones XEP-0163[43] y XEP-OMEMO[44] para su correcta implementación.

OMEMO se diferencia de OTR tiene la capacidad de utilizar múltiples dispositivos, conversaciones grupales cifradas e iniciar chats de forma asincrónica. Por su parte OTR tiene la ventaja de que se puede añadir cifrado entre extremos a cualquier protocolo de chat.

5.4 Ricochet

Ricochet es un sistema de chat donde la comunicación es anónima y cifrada entre extremos. En su sitio web se destacan las siguientes características: no existen metadatos que puedan ser espiados por un tercero; no se puede saber a quién pertenece una cuenta, salvo que se identifique; existe anonimato porque no se sabe la ubicación de los participantes de la comunicación; por último no se requiere configurar nada para que el chat sea secreto y anónimo.[92]

Cada cliente de manera transparente implementa un servicio oculto de Tor y el identificador en lugar de ser un número telefónico o un nombre de usuario, es el *hostname* de la dirección del servicio oculto que se escribe de la siguiente manera: ricochet:sa3usfhhca7kkhgp. Es por esto que la

42 Existe un listado con servidores que lo soportan https://compliance.conversations.im/
43 https://xmpp.org/extensions/xep-0163.html
44 https://conversations.im/omemo/xep-omemo.html

comunicación es siempre autenticada. Sin embargo existe el inconveniente de que los nombres no son fáciles de recordar.

Otra característica a destacar es que la comunicación es P2P y que el chat se realiza directamente entre clientes sin la necesidad de un servidor intermedio como se puede ver en la imagen 26. Con la cual se elimina el riesgo de tener un administrador que pueda monitorear el tráfico.

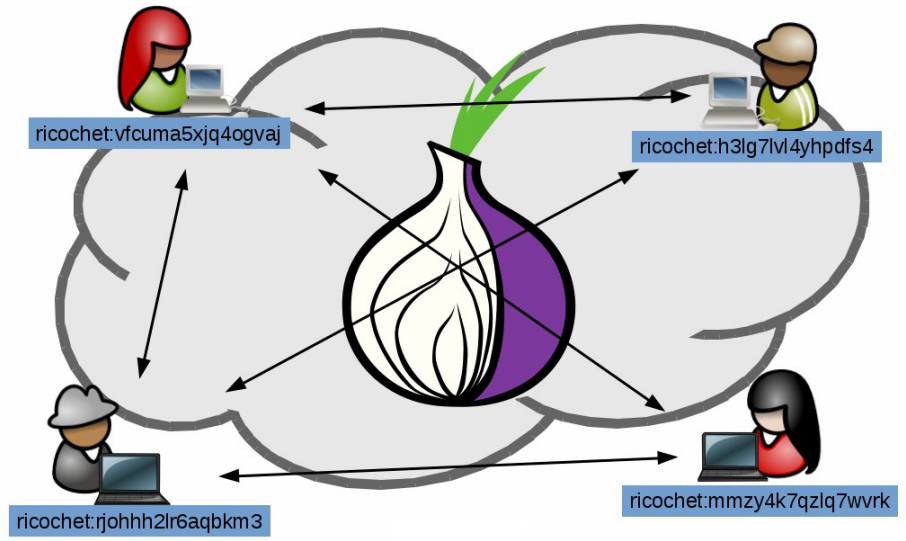

Imagen 26: Ricochet cliente P2P
Fuente. Elaboración propia con gráficos de Openclipart y logo de Tor

Más allá de que los nombres no sean fáciles de recordar la Interfaz de uso de Ricochet es simple y recuerda a la de cualquier cliente de mensajería instantánea. Una vez que se añade un contacto se pone un alias fácil de identificar para saber con quién se esta chateando. En la imagen 27 se puede ver la Interfaz de lista de contactos junto a una conversación.

Ricochet es bueno para tener secreto y anonimato en las comunicaciones sin mayor interacción por parte del usuario. Se debe considerar que no se puede tener conversaciones asincrónicas ni grupales; además no existen clientes móviles.

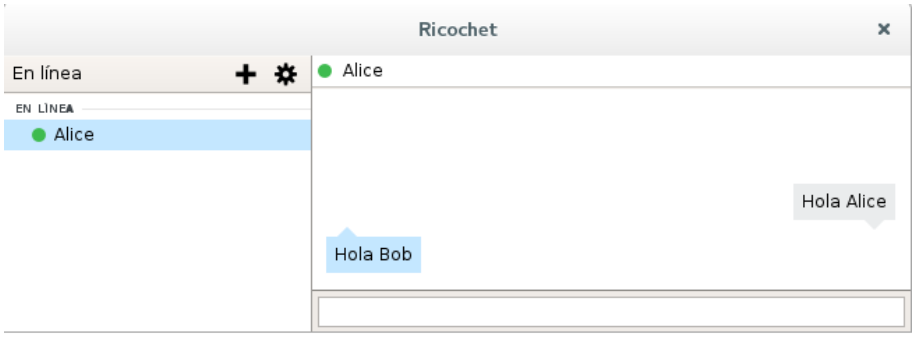

Imagen 27: Captura de pantalla Ricochet
Fuente: Elaboración propia.

Otra tema a considerar es que la última versión estable al momento de escribir este documento es del 5 de noviembre de 2016.[92] Al no tener un desarrollo activo es probable que tenga fallas de seguridad. Sin embargo el concepto de un medio de comunicación cifrado entre extremos que no requiere servidores es algo que se debe tomar en cuenta.

5.5 Briar

Briar es una aplicación libre para Android que provee mensajería instantánea y otros servicios de manera descentralizada sin la necesidad de servidores. Al igual que Ricochet, Briar utiliza servicios ocultos de Tor para permitir la comunicación entre las partes. Adicionalmente es posible comunicarse a través de la red Wifi, Bluetooth, e incluso se tiene planificado permitir sincronizar las comunicaciones a través de medios alternativos como memorias USBs u otros.[93]

Según el manual de Briar[94] y pruebas realizadas se explicará el funcionamiento de este sistema. Al crear una cuenta se pide un alias y una contraseña. La contraseña sirve par cifrar la cuenta y todo se guarda de manera local en el teléfono; no se crea un registro en ningún servidor. Por este motivo la información no es accesible por terceros pero si se extravía el teléfono o se olvida la contraseña se pierde el acceso a la cuenta y hay que empezar de nuevo.

Existen dos formas en las que se puede añadir contactos en Briar. La primera es en persona, en este caso Alice y Bob se reúnen y al utilizar de manera conjunta los códigos QR y la conexión de Bluetooth del celular

intercambian las llaves públicas. Es así como Alice y Bob autentican su cuenta al momento de presentarse y evitan ataques de hombre en el medio.

Si bien esta medida limita la capacidad de un ataque de hombre en el medio, genera un problema a la hora de añadir contactos ya que se deben encontrar en persona. Como opción alternativa Briar se puede presentar contactos. Si Alice conoce a Bob y a Carol los puede presentar para que ellos puedan comunicarse a través de Briar como se puede ver en la imagen 28.

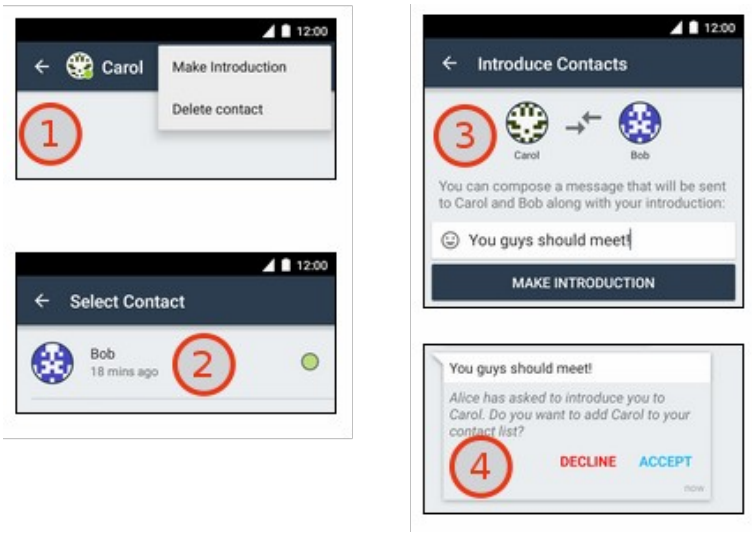

Imagen 28: Presentación de contactos en Briar
Fuente: Elaboración propia mediante gráficos proyecto Briar

Torsten Grote, desarrollador de Briar, explica en su conferencia sobre esta aplicación en el CCC[45] 2017 que esta es la forma en la que funciona la sociedad. Lo normal es que seamos presentados por las personas que conocemos. Se conocen a otras personas porque esas personas son presentados por amigos, familiares o cualquier otro conocido. [93]

Una característica interesante descrita por Grote es que Briar puede añadir distintas formas de comunicación. Es por esto que se puede enviar mensajes a través de Tor, la red local o Bluetooth. Se está trabajando para que incluso se pueda enviar mensajes de manera asincrónica a través de memorias USB. Bromea Grote, en la conferencia, que se podría hacer cosas

45 Chaos Communication Congress es un evento de seguridad informática que se realiza en Berlin desde 1984.

como enviar mensajes cifrados en una memoria USB utilizando una paloma mensajera.

Al igual que Signal y OTR se puede tener secreto adelante. Sin embargo como el sistema considera importante que los mensajes puedan viajar de forma asincrónica con demoras que en unos casos pueden durar minutos y en otros incluso días; se decidió no negociar una clave nueva en cada mensaje sino que las claves caducan con un valor de tiempo fijo que lo puede configurar el usuario.

5.6 Delta Chat

Delta chat es un cliente de mensajería instantánea para celular que funciona a través de cuentas de correo electrónico para enviar y recibir mensajes; funciona con OpenPGP de manera transparente. [95]

Para el intercambio de llaves utiliza Autocrypt que es un estándar para hacer accesible el uso de correo cifrado con OpenPGP. Este estándar se encuentra en desarrollo y se divide en tres niveles de los cuales el primero es el único implementado. En este nivel las llaves públicas se comparten automáticamente en todos los mensajes por lo que empezar a cifrar las comunicaciones es sencillo.[96] [97]

Delta Chat permite autenticar las partes verificando manualmente las llaves públicas como se puede ver en la imagen 29. Esta validación es similar a como lo harían aplicaciones de chat como Signal.

El uso OpenPGP permite tener comunicaciones asincrónicas sacrificando la propiedad de secreto perfecto. Tiene la ventaja sobre herramientas como Signal de no requerir un identificador fuerte como el número de teléfono. Todo el funcionamiento de correo electrónico visto en el capítulo 3 funciona con Delta Chat. Es decir se pueden crear cuentas anónimas en servidores públicos o usar un servidor de chat publicado en un servicio oculto.

Imagen 29: Verificación de llaves con Delta Chat
Fuente: Captura de pantalla

Más allá del anonimato, Delta Chat es útil para organizaciones que quieren ganar independencia en las comunicaciones y que ya cuentan con un servidor de correo electrónico que funcione con IMAP seguro. En este caso basta con instalar Delta Chat en los celulares y tendrán mensajería instantánea de manera similar al uso que se le da hoy en día a Whatsapp.

5.7 Análisis Chat

Es posible tener comunicaciones secretas de chat con la combinación de Tor, XMPP y OTR. Esta no es una única solución y por eso se presentó un análisis adicional de otras opciones con sus ventajas y desventajas.

Se eligió XMPP por ser un protocolo estándar y tener varios clientes soportados. Este protocolo permite federar y además es fácil de implementar como un servicio cebolla. Para el cifrado de los mensajes se puede elegir entre OTR y OMEMO. OMEMO es menos seguro y más versátil porque permite sincronizar las conversaciones entre varios dispositivos.

Briar y Ricochet tienen la característica de no requerir servidores centralizados. Para que Ricochet funcione se requiere que las dos personas estén conectadas a la vez por lo que no se puede tener conversaciones asincrónicas.

Briar funciona de una manera diferente al resto ya que para añadir un contacto hay que encontrarse en persona o deben ser presentados por

un contacto en común. Esto puede limitar que tenga una base grande de usuarios de forma rápida. Por otro lado organizaciones de personas que trabajan juntos podría utilizarlo fácilmente sin necesidad de infraestructura de servidores.

Delta Chat tiene la ventaja de que puede utilizar infraestructura existente, permite tener conversaciones grupales y asincrónicas. Tiene la desventaja de no soportar secreto perfecto hacia adelante ni repudio.

6 Llamadas de Voz sobre IP Secretas

Con herramientas como Skype, Whatsapp, Signal o Telegram se pueden realizar llamadas a través de Internet de forma sencilla y cifrada. En el capítulo 1 se explicó que Microsoft y Facebook, las empresas dueñas de Skype y Whatsapp respectivamente, son miembros del programa PRISM de la NSA según documentos de Snowden. Si bien Signal y Telegram no se conoce que tengan relación con agencias de inteligencia y permiten tener llamadas cifradas los metadatos de las mismas podrían estar siendo vigilados y almacenados.

Cómo se comentó en el capítulo anterior otro problema de aplicaciones como Telegram o Signal es que requieren de un número telefónico para crear las cuentas.

Existen protocolos abiertos como XMPP con la extensión Jingle[46] o SIP que permiten tener comunicaciones de voz sobre IP. Los mismos funcionan con una arquitectura cliente servidor para gestionar las cuentas y la señalización de las comunicaciones. En la sección 6.1 se analizará como proteger el protocolo SIP utilizando los protocolos SRTP en combinación ZRTP.

En la sección 6.2 se analizarán protocolos que descentralizan la gestión de cuentas y la señalización de las comunicaciones para poder tener comunicación P2P. Sobre los mismos se verá si es posible comunicarse de forma secreta.

Por último en la sección 6.3 se mostrará el uso de la plataforma Mumble para tener reuniones virtuales en las que los miembros de las mismas son los únicos que pueden saber de que se habló y que la conversación en efecto sucedió.

46 Extensión para el protocolo XMPP que permite tener comunicaciones de voz.

6.1 Cifrado Extremo a Extremo y Voz sobre IP

Varios protocolos de voz sobre IP separan la señalización de la comunicación con el transporte del contenido de la misma. Le señalización consiste en los pasos necesarios para establecer y terminar la llamada; es decir quién llama, quién contesta, etcétera. El transporte de la comunicación es el audio y el video que se utiliza para que la misma suceda.

El Protocolo de Señal de Inicialización, SIP por sus siglas en Inglés, es un estándar que existe desde 2002 con el RFC 3261. Se encarga de establecer la comunicación entre dos o más personas; enviar el mensaje para que el teléfono timbre; identificar direcciones IPs de los miembros de la comunicación. Funciona tanto en TCP como UDP en el puerto 5060 y en el puerto 5061 de TCP para TLS.[98]

Para el contenido de la comunicación se utiliza el Protocolo de Transporte en Tiempo Real, RTP por sus siglas en inglés. Este es un protocolo que está pensado para la transmisión de audio, video u otro tipo de información que requiera llegar en tiempo real. La comunicación se la realiza directamente entre las partes según la señalización enviada por protocolos como SIP;[99] no es necesario que la comunicación pase a través del servidor SIP.

Este esquema funciona bien en redes LAN pero tiene problemas cuando se quieren establecer comunicaciones a través de Internet donde los participantes no tienen direcciones IP públicas. En este caso se utiliza el protocolo STUN que está pensado para solventar problemas relacionados con NAT en protocolos como SIP combinado con RTP.[100]

La comunicación a través RTP no va cifrada por defecto. Para cifrar entre extremos se puede utilizar una combinación entre los protocolos SRTP y ZRTP. El primero se encarga de cifrar las comunicaciones y en su especificación explican que de manera mandatoria debe soportar AES, aunque se podría utilizar otros algoritmos de cifrado.[101]

El protocolo ZRTP se encarga del intercambio de las llaves que serán utilizadas en SRTP. Funciona de manera descentralizada y negocia la llave de cifrado a través de una negociación Diffie Hellman efímera. En cada sesión se negocia una llave simétrica de cifrado por lo que este protocolo tiene la característica de secreto perfecto hacia adelante. Para protegerse de un posible ataque de hombre en el medio se utiliza una Cadena de Autenticación Pequeña, SAS por sus siglas en inglés. Esta cadena es presentada a las personas que intervienen en la comunicación en la pantalla de su teléfono de software o hardware y la comparan hablando. Si las cadenas coinciden, la probabilidad de un ataque de hombre en el medio es baja.[102]

Se explicó el uso de ZRTP para el protocolo SIP con SRTP. Es importante notar que ZRTP también puede ser utilizado con el protocolo XMPP que soporte la extensión Jingle.[103] De esta manera se podría tener un mismo servidor chat que también soporte voz sobre IP.

Sea cual fuere el caso, el uso de SRTP requiere comunicaciones a través de voz sobre IP que funcionan sobre el protocolo UDP. Esto es un problema para ocultar el tráfico a través de Tor ya que esta red solamente soporta TCP. Una alternativa podría ser utilizar I2P en lugar de Tor, sin embargo no se hicieron pruebas al respecto.

Onioncat es un adaptador de VPN para Tor e I2P que brinda la posibilidad de hacer accesible un equipo a través de una dirección IPv6 publicada como servicio oculto. De esta manera dos o más máquinas que tengan configurado Onioncat podrán comunicarse entre sí como si estuvieran conectadas al mismo *switch*. Esto incluso podría permitir establecer comunicaciones de voz sobre IP. [104]

El protocolo SIP puede funcionar sin la necesidad de un servidor intermedio y realizar llamadas directas entre dos dispositivos si los mismos sen encuentran en la misma red LAN o si los dos tienen IP públicas. Basta con realizar la llamada de la forma sip:<nombrehost o dirección IP>. [105] Durante las pruebas de laboratorio se planteó la posibilidad de utilizar

Onioncat para llamadas de voz sobre IP directamente entre dos clientes SIP, para esto se utilizó el cliente libre Linphone.

Se logró con éxito realizar llamadas dentro de la red LAN, sin embargo no fue posible hacerlo utilizando IPV6 en combinación con Onioncat. Esto no quiere decir que no se pueda hacer, se pararon las pruebas porque se encontraron métodos que funcionaron de manera más fácil según los conocimientos técnicos del autor. Sin embargo sí se logró hacer ping6 entre dos máquinas ubicadas de forma remota como se puede ver en la imagen 30.

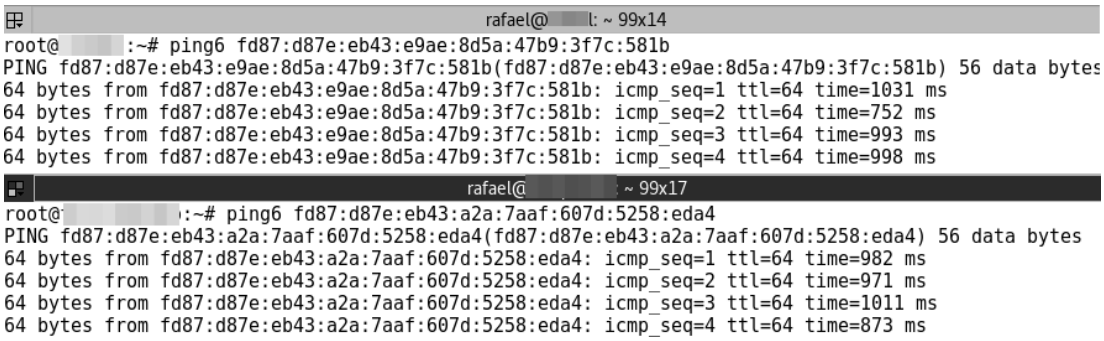

Imagen 30: Ping6 con Onioncat
Fuente: Elaboración propia en base a capturas de pantalla.

Esto demuestra que es posible conectar dos máquinas remotas como si estuvieran juntas a través de IPv6 utilizando Onioncat por lo que debería ser posible hacerlo para voz sobre IP.

6.2 Tox y Ring

Existen varios proyectos que permiten añadir una capa de cifrado a protocolos de voz y video como son SIP y XMPP con Jingle. Entre ellos se puede mencionar a Jitsi o a Linphone. Sin bien con estas herramientas se puede tener secreto en las comunicaciones, no permiten ocultar el hecho de que la comunicación sucedió.

Proyectos como Tox[47] o GNU Ring[48] eliminan la necesidad de tener un servidor central para gestionar las cuentas de las personas. Para eso

47 Protocolo de comunicaciones descentralizado con varios clientes disponibles https://tox.chat/
48 Proyecto descentralizado de comunicaciones https://ring.cx/

utilizan una base de datos distribuida de hash (DHT) donde se registran las cuentas de las personas. Tener una base de datos distribuida de los usuarios podría poner en riesgo el anonimato ya que algunos metadatos podrían estar disponibles para los nodos de la red.

GNU Ring es una aplicación que soporta el protocolo SIP estándar pero además realizó una modificación sobre el mismo para funcionar de manera descentralizada. Los usuarios se registran en una base de *hash* descentralizada utilizando el protocolo Kademlia. Esta base permite identificar las direcciones IP asociados a un identificador Ring y de esta manera se establece la comunicación entre los mismos. [106] [107]

Se utiliza SRTP para las comunicaciones de audio y video donde se intercambia las llaves a través de Diffie Hellman. El funcionamiento es similar al de ZRTP que lo dejaron de utilizar algún momento antes de agosto de 2016. [108]

El proyecto Ring es interesante por ser descentralizado sin embargo no garantiza el anonimato de las comunicaciones. En su página web se explica que al depender de una base de datos distribuida como OpenDHT recolecta y guarda metadatos para su funcionamiento. Esto podría hacer que un espía vigile uno o varios nodos de la red DHT y así saber quien está hablando con quien o al menos sus direcciones IP.[109]

Tox por su parte es un protocolo nuevo que busca crear comunicaciones seguras de manera descentralizada. Al ser un protocolo no se trata de una aplicación sino que existen varios clientes disponibles para escritorio y telefónica móvil.[110]

Al igual que Ring, Tox utiliza OpenDHT para gestionar las comunicaciones P2P. Se utiliza un *hash* de la llave pública del usuario como identificado del mismo. Por defecto Tox no intenta ocultar la dirección IP de las personas ya que esta es necesaria para que dos personas puedan conversar entre sí. A diferencia de Ring, Tox puede funcionar con TCP por lo que soporta el uso de Tor y así se podría ocultar la dirección IP. [111]

De esta manera Alice y Bob podrían comunicarse entre sí, sin revelar su actividad a terceros. Se realizaron pruebas con el cliente Trifa para Android en combinación con Orbot para utilizar Tor. Se crearon las cuentas a través de Tor pero no fue posible añadir contactos utilizando exclusivamente Tor. Al igual que en el caso de SIP con Onioncat esto no quiere decir que no se pueda hacer, pero se encontró una solución sencilla que se verá en la siguiente sección.

6.3 Comunicaciones Secretas con Mumble

Mumble es una aplicación cliente servidor creada para la transmisión de audio en grupo. Originalmente se desarrollo como herramienta de comunicación para video juegos en red, sin embargo también se la utiliza para gestionar reuniones de forma virtual. Tiene la característica de estar optimizada para utilizar poco ancho de banda y funcionar bien con baja latencia; por lo que se desempeña adecuadamente con malas conexiones a Internet; además puede funcionar a través de TCP. [112] Esta última característica resulta conveniente para ser utilizado con Tor.

Mumble, está compuesto por el cliente que lleva ese nombre y el servidor conocido como Murmut. La comunicación en Mumble siempre está cifrada entre clientes y el servidor. El canal de control que se encarga de comunicar los mensajes de chat y otra información que no requiera entrega inmediata se envían a través de TLS con AES 256 bits. La comunicación en tiempo real se cifra OCB-AES con 128bits.

Se utilizan certificados para autenticar al servidor y de manera opcional a los clientes. Los certificados del servidor son auto firmados de manera predeterminada por lo que se deben verificar manualmente la primera vez que se usan. En la imagen 31 se puede ver el esquema de normal de Mumble donde dos o más personas se comunican y el tráfico va cifrado entre clientes y servidor. El administrador del servidor podría espiar las conversaciones y metadatos de la comunicación. En este caso Alice y

Carlos utilizan el cliente de escritorio en su computadora y Bob el cliente desde un teléfono móvil.

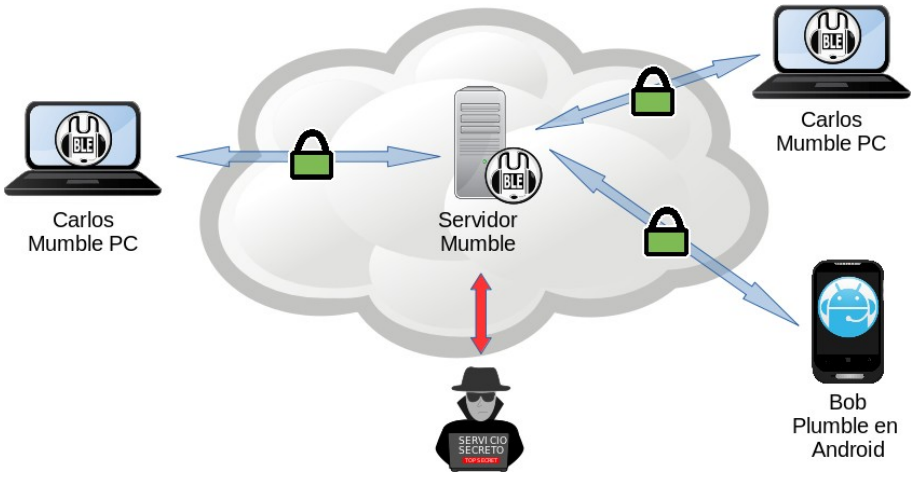

Imagen 31: Imagen 31: Esquema de Mumble sin Tor
Fuente: Elaboración propia con imágenes de Openclipart y logos de Mumble y Plumble

Para eliminar la necesidad de tener que confiar en que el administrador del servidor no espíe las comunicaciones se puede publicar Murmut como un servicio cebolla. Alice, en su computadora, puede ejecutar tanto el servidor de Mumble como el cliente y ser ella quien gestione la comunicación como se puede ver en la imagen 32.

Al utilizar el servicio oculto de Tor ya no es necesario tener una dirección IP pública por lo que Alice puede ejecutar el servidor en su propia computadora o un servidor que ella controle. Los tres establecen la conexión hasta el servidor y pueden tener una conversación en línea.

Adicionalmente a la capa de cifrado de Mumble se tiene una capa de cifrado de Tor. Esta capa autentica el canal desde los clientes hasta el servicio oculto por lo que si bien la primera vez que alguien se conecta a Mumble recibirá una advertencia del certificado, esta se podrá ignorar ya que el canal está autenticado gracias a Tor.

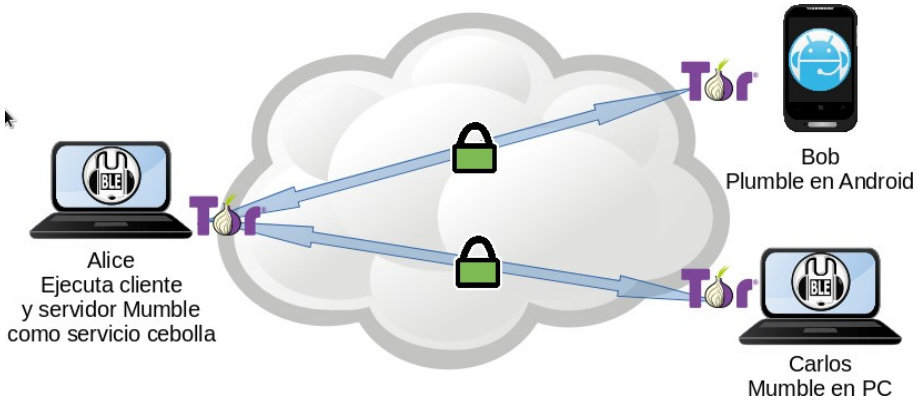

Imagen 32: Mumble como servicio cebolla
Fuente: Elaboración propia con imágenes de Openclipart y logotipos de Mumble, Plumble y Tor

Durante el desarrollo de este trabajo, el autor gestionó reuniones virtuales con participantes en cuatro países diferentes mediante esta solución. La calidad del audio funcionó bien, aunque es importante que una sola persona hable a la vez; este problema también se puede tener en otras soluciones de conferencia.

Hay que tomar en cuenta que la persona que gestiona la comunicación, Alice en este ejemplo, debe considerar el ancho de banda de subida. Por cada persona que se sume a la sesión será un canal de comunicación adicional.

La instalación del Murmut en Linux es sencilla ya que el paquete está disponible en los repositorios de la mayoría de las distribuciones. La configuración básica de las pruebas consistieron en limitar las conexiones a 127.0.0.1 para que no sea visible en la red local. Adicionalmente se puso una contraseña para acceder a la sala de chat. Para esto se añadieron las siguientes líneas en el archivo /etc/mumble.ini:

```
host=127.0.0.1
[…]
serverpassword="ClaveSecreta"
```

Se configuró Tor para publicar el puerto 64738 de 127.0.0.1 como servicio cebolla:

```
#Mumble
HiddenServiceDir /var/lib/tor/mumble
HiddenServicePort 64738 127.0.0.1:64738
```

Se realizaron las pruebas desde celular utilizando Plumble combinado con Orbot y desde el sistema operativo Linux se utilizó el comando torify para enviar el tráfico de Mumble[49] por Tor. En la imagen 32 se puede ver a la izquierda la configuración del cliente de celular Plumble y a la derecha el cliente para escritorio Mumble.

Imagen 33: Configuración clientes Mumble
Fuente: Elaboración propia a partir de capturas de pantallas.

Con poco esfuerzo se logró tener conversaciones secretas a través de Internet, sin embargo se requiere conocimiento técnico de Linux. Tecnologías como máquinas virtuales o contenedores tipo Docker permitirían tener un servidor Mumble a personas que no tienen el conocimiento técnico para hacerlo. Si Alice quiere comunicarse con Bob bastaría con hacer doble click en su escritorio y se levanta la máquina virtual lista para proveer el servicio.

6.4 Análisis de Llamadas Secretas

En un inicio se pensó que tener comunicaciones secretas de voz sobre IP sería imposible. Se sabía que el cifrado entre extremos es algo que funciona bien desde hace algún tiempo. En lo que se dudaba es si era posible utilizar tecnologías como Tor para comunicaciones habladas en

49 Junto a la Fundación Acceso de Costa Rica el autor desarrollo un manual para instalar Mumble como servicio oculto de Tor: https://medium.com/@facceso.ca/comunicaciones-secretas-y-autónomas-de-voz-ip-con-mumble-3b3d3bad06c5

tiempo real. El autor creía que a lo mucho se podría tener comunicaciones de voz sobre IP utilizando tecnologías como VPN.

Ring y Tox permiten tener comunicaciones descentralizadas con cifrado entre extremos de manera mandatoria. El problema de estas soluciones es que al utilizar una base de datos distribuida de *hashes* los nodos de la misma podrían extraer metadatos. Una ventaja de Tox sobre Ring es que se puede utilizar con Tor y de esta manera tener anonimato con respecto a los nodos DHT.

La solución de Mumble no deja huella por fuera de los dispositivos de las personas que participan en las comunicaciones. Implementar el servicio oculto es simple pero se requiere conocimientos básicos de Linux. El uso de máquinas virtuales o contenedores como Docker podrían simplificar esta tarea.

Si bien no funcionó en los laboratorios la opción de conectar dos teléfonos de software que soporten SIP con IPV6 a través de Onioncat, es una opción a tomar en cuenta. Si se tuviera teléfonos de software que lo hagan de forma transparente se tendría algo similar a Ricochet pero para conversaciones habladas.

7 Aportes Originales del Autor

La tecnología tiene un impacto político en la sociedad. La privacidad esta vulnerada por la vigilancia masiva en la red; sin embargo se la puede proteger. Un primer aporte de este trabajo es mostrar cuan vulnerado se encuentra el derecho humano a la privacidad y que a pesar de esto es posible tener comunicaciones secretas.

La vigilancia global de la NSA y la estatal en América Latina son ejemplos del uso de la tecnología para controlar. La concentración de datos personales y el monitoreo de las comunicaciones de miles de millones de personas es una fuente de poder en manos de pocas empresas. Este poder puede ser compartido con agencias de inteligencia como se vio en el caso de PRISM.

En la otra orilla existen herramientas tecnológicas que ayudan a proteger la privacidad. El software libre combinado con criptografía permiten defender el derecho humano a la privacidad. El software libre a más de poder ser auditado, es accesible y no pone barreras económicas para su uso. La criptografía permite, a través de las matemáticas, que algo que deba ser secreto pueda serlo.

De esta manera, un grupo de personas podrá comunicarse de forma segura utilizando software libre y hardware de pocos recursos. Podrían utilizar un computador viejo, máquinas virtuales o hardware de bajo costo para implementar su propio servidor de chat, correo o Mumble como servicio cebolla; y así tener comunicaciones seguras y autónomas para organizarse.

No importa si se trata de organizaciones de defensores de derechos humanos, periodistas o cualquier otra; la comunicación es un recurso estratégico. Es por esto que durante este trabajo se evita utilizar servicios centralizados y se busca la autonomía.

Más allá de las herramientas tecnológicas, la seguridad de las comunicaciones implica tener confianza en las personas. Alice debe confiar

que Bob no lo va a traicionar, pero muchas veces Alice y Bob deberán confiar en quién administra los servidores de comunicaciones. Por esto en este trabajo se plantea tener servidores gestionados como servicios ocultos. Si se va a confiar en el administrador del sistema, mejor que sea alguien conocido.

*****

En el año 2013 tras las revelaciones de Snowden se incrementó la conciencia política sobre la importancia de la privacidad en Internet. Desde el punto de vista tecnológico se empezó a masificar el uso del cifrado entre extremos. El caso más significativo fue el de Whatsapp que en 2016 dio la posibilidad a mil millones de personas de cifrar sus comunicaciones entre extremos.

Asumiendo que Whatsapp no pueda leer el contenido de las comunicaciones, sí puedo leer los metadatos de las mismas. Además esta herramienta utiliza los números telefónicos para autenticar a las personas lo que es un problema grave desde el punto de vista de anonimato.

Un aporte importante de este trabajo en la era pos Snowden es mostrar que no es suficiente con ocultar el contenido de las comunicaciones, sino que hay que ocultar el hecho de que las mismas ocurrieron. Es por esto que se plantea la opción que ocultar los metadatos con redes de anonimato y el contenido de las comunicaciones con cifrado entre extremos. El autor no es el primero en proponer este tipo de soluciones, sin embargo en este trabajo se hace una extensa investigación de las soluciones existentes con un análisis crítico.

La solución para la pérdida de la privacidad en Internet no consiste en instalar herramientas mágicas. Es necesario tener una visión crítica de la tecnología y cuestionar las soluciones existentes. Se deben buscar las fallas de seguridad para poder corregirlas. Es por esto que este documento se encarga de analizar varias opciones de comunicaciones seguras. Se invita al lector a tener ojo crítico con las herramientas presentadas, pero también con este trabajo.

*****

El desarrollo de este trabajo no se limitó a documentar las herramientas existentes. En casi todos los casos se hicieron pruebas de funcionamiento prácticas; pero también se dictaron talleres y se hicieron laboratorios con gente interesada en la privacidad. Parte importante del conocimiento adquirido para desarrollar este trabajo se lo ha conseguido al compartir y aprender con otros.

Un ejemplo importante fue el taller de 15 horas sobre Comunicaciones Seguras desarrollado por el autor y dictado por él en los meses de octubre y noviembre de 2017 como taller opcional en la Maestría de Seguridad Informática de la Universidad de Buenos Aires. En el mismo se enseñó a crear cuentas anónimas de correo electrónico y de chat utilizando cifrado entre extremos; se hicieron pruebas de uso para voz sobre IP con GNU Ring; y otros temas de seguridad digital que no se trataron en este trabajo.

A inicios de 2018, el autor en conjunto con miembros de la comunidad de software libre de Buenos Aires crearon la iniciativa Privacidad Global[50]. La misma busca masificar la conciencia sobre la importancia de la privacidad en el mundo digital. Como parte de la experiencia con Privacidad Global se creó un foro anónimo como servicio cebolla para preguntas y respuestas relacionadas con privacidad y anonimato. Adicionalmente se realizaron pruebas de tecnologías como Matrix, Retroshare y Delta Chat. Al momento de escribir este documento, el autor sigue participando en un grupo de Delta Chat de Privacidad Global.

De regreso a Ecuador, el autor empezó a colaborar con la Fundación Acceso de Centro América. Esta organización brinda apoyo tecnológico en seguridad digital a defensores de derechos humanos en países como El Salvador, Guatemala, Honduras y Nicaragua. Parte importante del trabajo realizado para la Fundación Acceso ha sido compartir conocimiento sobre comunicaciones seguras.

50 https://privacidad.global/

El autor utilizó un servicio oculto de Tor con Mumble alojado en su casa para organizar la agenda y metodología de un encuentro de seguridad digital en Guatemala. Se realizaron varias reuniones virtuales donde se interactuó con gente de Nicaragua, Guatemala, Honduras y el Salvador de manera simultánea. Como resultado exitoso de estas pruebas se solicitó al autor que escriba un tutorial sobre su funcionamiento, el mismo se publicó en el blog de la organización.[51]

En noviembre de 2018 el autor dictó un taller sobre servicios cebolla en el encuentro de Guatemala. En el mismo se enseñó a tener autonomía en comunicaciones de voz sobre IP con Mumble y a compartir archivos de forma segura con Onionshare y Nextcloud; estas últimas son herramientas alternativas a sistemas como Dropbox.

Estas experiencias sirvieron para entender el funcionamiento de muchas de las herramientas presentadas en este trabajo pero también para que la gente aprenda a utilizarlas y protejan sus comunicaciones con las mismas. El autor considera que la privacidad es importante para todos, pero hay gente que la necesita más. América Latina es una región donde se ha abusado los derechos humanos de las personas por lo que se espera que este trabajo sirva para los que más lo necesitan.

51 https://medium.com/@facceso.ca/comunicaciones-secretas-y-autónomas-de-voz-ip-con-mumble-3b3d3bad06c5

Conclusiones

Las comunicaciones en Internet son vigiladas por grandes potencias como los Estados Unidos y sus aliados o por los gobiernos locales. Estos son capaces de espiar las comunicaciones de forma masiva a todos los ciudadanos de un país o a escala global como lo hace la NSA.

En este trabajo se describió la vigilancia masiva y la vigilancia dirigida. La primera intenta espiar a una gran parte de la población de forma recurrente. Un ejemplo claro de esto podría ser espiar todas las comunicaciones de Internet de un país. La vigilancia dirigida es donde un Estado u otro ente intenta vigilar a una persona en particular. Una forma de hacerlo es instalar *malware* en teléfonos o computadoras como lo hacen empresas como Hacking Team, NSO o Fin Fisher.

El modelo de amenaza que se intenta proteger en este trabajo contempla dos tipos de adversarios. El primero es el proveedor de Internet, gobierno o cualquiera que pueda espiar el tráfico de Internet de un usuario. El segundo es el proveedor de un servicio que podría espiar los metadatos y los datos de sus usuarios.

No se analizó la forma de defender las estaciones de trabajo ante posibles ataques de software malicioso espía. Si la computadora o teléfono celular está infectado con algún *malware*, se puede romper la seguridad de las comunicaciones. Se invita al lector a investigar sobre las distribuciones Linux pensadas en seguridad y privacidad como Tails, Whonix o Qubes.

La primera capa de protección que se vio son las redes de anonimato y VPNs. Cuando se usa VPNs se transfiere la capacidad que tiene el proveedor local de Internet de espiar al proveedor de VPN. El proveedor local no podrá saber que hace el usuario, pero sí sabrá que está usando VPN y es posible que el adversario contacte a ese proveedor. El administrador del servicio al que el usuario se está conectando sabrá que el tráfico viene de una VPN. El proveedor de VPN podrá monitorear todo el tráfico del usuario por lo que se debe tener mucho cuidado al escoger un

proveedor de este tipo, ya sea este comercial, gratuito o provista por algún conocido.

La red Tor por su parte permite brindar anonimato en relación con el proveedor de Internet así como del proveedor del servicio. A diferencia de VPNs en este caso no se confía los registros de todo el tráfico a un tercero. Adicionalmente con Tor se pueden crear los servicios cebolla que permiten proveer sistemas de manera anónima y autónoma.

Una tercera opción es I2P que tiene la característica de ser una red más descentralizada que Tor. I2P permite publicar servicios ocultos tanto TCP como UDP pero no está pensada para acceder a Internet convencional de forma anónima.

Se decidió hacer las pruebas con Tor ya que permite trabajar tanto en la red clara como en redes obscuras. De esta forma se pueden lograr comunicaciones secretas en servidores de terceros o gestionar servicios propios mediante servicios cebollas.

Al utilizar Tor correctamente se mitiga el riesgo de que tanto el proveedor del servicio como el proveedor local de Internet sepan quien se comunica con quien. Sin embargo el proveedor del servicio podría leer el contenido de las comunicaciones y a través del contenido de las mismas saber quien se comunica con quien.

Es por esto que el segundo paso para las comunicaciones secretas es usar el cifrado entre extremos. Si Alice y Bob crean cuentas de forma anónima, acceden a las mismas siempre de forma anónima y cifran sus mensajes entre extremos; las comunicaciones pueden ser secretas.

Para tener mayor autonomía se puede gestionar infraestructura propia mediante servicios ocultos. En el mejor de los casos los servicios están administrados por alguno de los participantes de la comunicación o en su defecto se debe buscar a alguien de confianza. Se logró tener comunicaciones exitosas con esta estrategia para correo electrónico, chat e incluso con voz sobre IP.

En el caso de correo electrónico se explicó como crear seudónimos para comunicarse de forma secreta. Para esto se utilizó el estándar OpenPGP en combinación con la red Tor. Adicionalmente se mostraron ejemplos en los que se utilizó un servicio de correo gestionado de manera autónoma como servicio cebolla.

Una debilidad del correo cifrado con respecto a los sistemas seguros de chat es que todos los mensajes se cifran con la misma clave. Esto quiere decir que si un atacante logra robar la llave privada de Alice podría descifrar los correos pasados y futuros de ella. Otra debilidad es que para autenticar un mensaje este debe ser firmado por lo que se tiene la característica de no repudio.

El caso de chat es más complejo ya que no existe un estándar dominante. Se decidió priorizar el uso de XMPP en combinación con OTR. Esto se debe a que XMPP es estándar maduro y que OTR ha demostrado funcionar para comunicaciones secretas.

Si bien se logró establecer comunicaciones secretas con esta combinación, se decidió explorar otras soluciones. Se analizaron opciones como el protocolo Signal y el caso particular de OMEMO en combinación con XMPP. En este análisis se vio que una ventaja importante del protocolo Signal es el soporte de conversaciones fuera de línea. Algo que ha sido incorporado en la versión 4 de OTR.

Se exploró brevemente el caso de Delta Chat. Esta herramienta no tiene la característica de repudio en comunicaciones autenticadas ni secreto perfecto hacia adelante como lo hacen OTR y los derivados de Signal. Por otro lado permite tener cifrado entre extremos de una forma rápida y sencilla sin invertir esfuerzo en infraestructura nueva ya que es una forma distinta de usar el correo electrónico.

Las herramientas Ricochet y Briar tienen en común la característica de no tener un servidor centralizado para las comunicaciones, cifrado extremo a extremo, autenticación y secreto hacia adelante. Todo esto viene con sacrificios en usabilidad. Ricochet requiere que las personas estén

conectadas de manera simultánea para poder conversar. Briar si bien permite tener comunicación asincrónica, estas no es tan transparentes como cuando se tiene un servidor conectado todo el tiempo; además requiere que los contactos sean añadidos en persona o mediante presentación.

Las comunicaciones de voz sobre IP es donde el autor tuvo mayor escepticismo. El cifrado entre extremos era algo que se sabía que funciona, comunicarse de forma anónima resultaba más complicado. Soluciones descentralizadas como GNU Ring o Tox parecían interesantes pero una base de *hash* distribuidos implicaba revelar metadatos a terceros. Si bien Tox dice que funciona con Tor no se logró tener pruebas exitosas.

Fue gracias a los comentarios en la lista de correo de Tor[52] que se evaluó la posibilidad de establecer reuniones secretas de voz en tiempo real a través de Mumble. Se realizaron pruebas con la gente de Privacidad Global para permitir la participación cuando alguien no pudo asistir en persona a una reunión. No fue hasta organizar las conferencias de audio con miembros de la Fundación Acceso para organizar el encuentro en Guatemala que se pudo ver que a más de ser un sistema seguro, este funciona bien y requiere poco ancho de banda.

La mayoría de las herramientas analizadas en este trabajo requieren conocimientos técnicos o cambios de hábitos; todas son software libre por lo que desde universidades, empresas, sociedad civil e incluso Estado se puede colaborar para hacer que las comunicaciones sean más seguras y fáciles de usar.

Es importante destacar que desde América Latina se está desarrollando tecnología que permite tener comunicaciones seguras. Por un lado se tiene OTR V4 que es liderado por gente de América del Sur; por otro lado se encuentra la investigación y desarrollo de criptografía post cuántica. Los dos casos muestran que cuando el conocimiento es accesible y se lo comparte, quiénes están en el sur son capaces de apropiarse de la tecnología y no ser simples consumidores.

52 Se puede leer el hilo de discusión acá: https://lists.torproject.org/pipermail/tor-talk/2018-May/044170.html

El derecho a la privacidad es un derecho humano que todas las personas merecen; sin embargo existe gente para la cual esta necesidad es más urgente que otros. No es lo mismo un disidente político en países como Venezuela o Nicaragua que un padre que quiere proteger la privacidad de su familia. Este trabajo será de mayor utilidad para quiénes tienen urgencia de proteger sus comunicaciones.

Esto no quiere decir que la privacidad no sea importante para el resto de personas. No se sabe que pueda pasar en el futuro por lo que la perdida de la privacidad hace vulnerable a las personas. Es importante generar conciencia en la sociedad sobre la importancia de proteger su vida personal. Solo así se podrá masificar el uso de tecnología que da control a las personas sobre sus datos y su vida en lugar de cederlo a corporaciones que basan su modelo de negocio en la extracción de datos personales.

Si se revisa la historia del siglo pasado en América Latina es importante recordar las dictaduras represivas que existieron. ¿Qué pasa si estas vuelven y los Estados tienen un aparato de vigilancia total donde pueden monitorear todas las comunicaciones de sus ciudadanos? Seguro será muy difícil escapar de gobiernos totalitarios.

“Las palabras se las lleva el viento” dice el proverbio. En el mundo físico sucede así y una conversación casual no deja registro, salvo que alguien explícitamente la grabe. En el mundo digital todo queda registrado, salvo que se haga algo al respecto. En este trabajo se mostraron varias estrategias para tener comunicaciones secretas y se espera que las mismas sean de utilidad para proteger a quiénes más lo necesitan.

Se viven los años dorados de la vigilancia masiva; tener comunicaciones secretas no es algo subversivo, es ejercer el derecho a la privacidad.

Anexo 1: Proveedores de Correo Electrónico

Este anexo se muestra un listado de proveedores de correo electrónico en los que se puede crear una cuenta de forma anónima a través de Tor sin necesidad de dar datos personales. En todos los casos se realizaron pruebas para validar el hecho de crear la cuenta de forma anónima. Es importante mencionar que en ningún caso se debe confiar la seguridad de las comunicaciones a un tercero.

Proveedor	Sitio web	Cifrado web	IMAP/POP	Comentarios
Protonmail	https://protonmail.com/ https://protonirockerxow.onion/	Sí, compatible con OpenPGP	Versión pago	Protonmail funciona con un sistema de cifrado incorporado en la interfaz web. Para crear la cuenta con Tor se debe resolver un captcha de Re-captcha o tener otra cuenta de correo electrónico no asociada. Es posible que no de ninguna de las dos opciones y que pida número de teléfono o donación. En ese caso se recomienda crear otro circuito de Tor en el navegador hasta tener la opción de captcha o correo alterno.
Tutanota	https://tutanota.com/	Sí	No	Tutanota tiene su propio sistema de cifrado y no es compatible con OpenPGP. Para crear una cuenta nueva se debe resolver un captcha propio de ellos y esperar de 24h a 48h para la activación de la cuenta.
Mailfence	https://mailfence.com/	Sí, compatible con OpenPGP	Versión pago	Permite crear una cuenta de forma anónima resolviendo un captcha propio y con una cuenta de correo para su activación. Si se desea tener anonimato se debe crear una cuenta alternativa en servicios como Disroot o Cock.li Mailfence permite utilizar OpenPGP desde la Interfaz web.
Disroot	https://disroot.org/	No	Sí	El único requisito para crear la cuenta a través de Tor es resolver Captchas de Re-Captcha. La cuenta también sirve para cuenta de XMPP, y otros servicios provistos por Disroot.
Cock.li	https://cock.li	No	Sí	Permite crear cuentas de forma anónima con Tor resolviendo un captcha sencilla, además permite escoger entre muchos dominios.

Anexo 2: Herramientas para Cifrado de Correo

En el presente anexo se describen algunas herramientas que se pueden utilizar para tener correo electrónico secreto. Hay que tomar en cuenta que no son las únicas. En este trabajo se realizaron las pruebas con Thunderbird, Enigmail y Torbirdy.

Herramientas	**Sitio web**	**Comentarios**
Thunderbird con Enigmail y Torbirdy	• https://www.thunderbird.net/es-ES/ • https://www.enigmail.net/ • https://trac.torproject.org/projects/tor/wiki/torbirdy	Thunderbird es un cliente de correo electrónico disponible para los sistemas operativos Windows, Linux y Mac compatible con los protocolos POP3 e IMAP seguros. Se le puede añadir los complementos Enigmail y Torbirdy. Enigmail permite tener soporte de OpenPGP para el correo electrónico y de esta manera cifrar, firmar y verificar la firma de correos, así como también la gestión de las claves. Torbirdy es un complemento que integra Thunderbird con la red Tor y provee otras medidas de seguridad para proteger la privacidad y el anonimato de los usuarios.
Mailpile	• https://www.mailpile.is/	Es un cliente de correo electrónico compatible con los protocolos: IMAPS, POP3S, SMTPS y SSH. A diferencia de clientes como Thunderbird, este corre un servidor web local al que se puede acceder a través de un navegador web. Para acceder al correo se debe abrir el navegador e ingresar a http://localhost:33411. La interfaz es similar a la de servicios como Gmail y permite hacer tareas como realizar búsquedas.
K9 con OpenKeychange	• https://k9mail.github.io/ • https://www.openkeychain.org/	K9 es un cliente de correo para Android compatible con POP3 e IMAP seguros. Para utilizar OpenPGP se lo suele combinar con la aplicación OpenKeychange.

Anexo 3: Clientes XMPP

En el presente anexo es un listado de clientes compatibles con XMPP y el soporte para OTR y OMEMO.

Aplicación	Web	Descripción	OTR	OMEMO
Pidgin	https://www.pidgin.im	Cliente de chat para sistemas operativos Windows y Linux. Tiene soporte para OTR y OMEMO a través de complementos. Se lo puede configurar para funcionar con Tor y es el cliente de chat disponible en la distribución Tails	X	X
Adium	https://adium.im/	Adium es un cliente de chat para Adium que utiliza la librería libpurple, la misma de Pidgin. Adium tiene soporte incorporado para OTR y puede soportar OMEMO a través de un complemento. Al igual que Pidgin se lo puede configurar para funcionar con Tor	X	X
CoyIM	https://coy.im/	CoyIM es un cliente de chat con soporte nativo para OTR y Tor. Adicionalmente tiene la opción incorporada para crear cuentas de chat a través de Tor.	X	
Conversations	https://conversations.im/	Conversations es un cliente de chat disponible para Android con soporte nativo para OMEMO. Tenía soporte para OTR pero este fue quitado en las últimas versiones.		X
Pix-Art	https://github.com/kriztan/Pix-Art-Messenger	Es un versión modificada de Conversations que entre cosas permite tener soporte para OTR.	X	X

Bibliografía

[1] R. Bonifaz, “La NSA Según las Revelaciones de Snowden”, Facultad de Ciencias Económicas. Universidad de Buenos Aires, 2017.

[2] “PRISM/US-984XN Overview”, abr-2013. [En línea]. Disponible en: https://edwardsnowden.com/2013/06/07/prism-overview-slides/.

[3] “NSA’s global interception network”, *Electrospaces.net*, 03-dic-2013. [En línea]. Disponible en: https://electrospaces.blogspot.com.ar/2013/12/nsas-global-interception-network.html. [Consultado: 19-jun-2017].

[4] B. M. Tecnología, “La poderosa herramienta de EE.UU. para vigilarlo todo en internet”, *BBC Mundo*, 08-ene-2013. [En línea]. Disponible en: http://www.bbc.com/mundo/noticias/2013/08/130801_tecnologia_snowden_nsa_xkeyscore_dp. [Consultado: 22-abr-2017].

[5] Wikileaks, “Vault7 - Home”, 07-mar-2017. [En línea]. Disponible en: https://wikileaks.org/ciav7p1/. [Consultado: 18-dic-2017].

[6] Wikileaks, “The Spy Files”, 01-dic-2011. [En línea]. Disponible en: https://wikileaks.org/the-spyfiles.html. [Consultado: 06-sep-2017].

[7] Wikileaks, “Spy Files - All Releases”, 2014. [En línea]. Disponible en: https://www.wikileaks.org/spyfiles/. [Consultado: 28-ago-2017].

[8] G. Pérez de Acha, “Hacking Team Malware para la Vigilancia en América Latina”. Derechos Digitales, mar-2016.

[9] Wikileaks, “Spy Files - Remote Control System: Full Intelligence on Target Users even for encrypted Communications”. [En línea]. Disponible en: https://www.wikileaks.org/spyfiles/document/hackingteam/287_remote-control-system-full-intelligence-on-target-users-even/. [Consultado: 24-ene-2018].

[10] Wikileaks, “Spy Files - Remote Control System”, oct-2011. [En línea]. Disponible en: https://www.wikileaks.org/spyfiles/document/hackingteam/147_remote-control-system/page-2/#pagination. [Consultado: 24-ene-2018].

[11] Animal Político, “México, el principal cliente de una empresa que vende software para espiar”, *Animal Político*, 07-jul-2015. .

[12] Citizen Lab, “The Million Dollar Dissident: NSO Group’s iPhone Zero-Days used against a UAE Human Rights Defender”, *The Citizen Lab*, 24-ago-2016. [En línea]. Disponible en: https://citizenlab.ca/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/. [Consultado: 29-dic-2017].

[13] R3D, Article 19, y SocialTIC, “Gobierno Espía”. jun-2017.

[14] Citizen Lab, “Mapping FinFisher’s Continuing Proliferation”, *The Citizen Lab*, 15-oct-2015. [En línea]. Disponible en: https://citizenlab.ca/2015/10/mapping-finfishers-continuing-proliferation/. [Consultado: 12-ene-2018].

[15] J. Scott-Railton, M. Marquis-Boire, C. Guarnieri, y M. Marschalek, “Packrat: Seven Years of a South American Threat Actor”, *The Citizen Lab*, 08-dic-2015. [En línea]. Disponible en: https://citizenlab.ca/2015/12/packrat-report/. [Consultado: 13-ene-2018].

[16] EFF, “¿Quiénes pueden vigilarnos? Infografía”, *Necessary and Proportionate*, 29-sep-2016. [En línea]. Disponible en: https://necessaryandproportionate.org/es/americas-reports/quienes-pueden-vigilarnos. [Consultado: 31-ene-2018].

[17] Privacy International, “Un estado en la sombra: vigilancia y orden público en Colombia”. ago-2015.

[18] “Our People – Moglen & Associates”. [En línea]. Disponible en: /people/. [Consultado: 22-ene-2018].

[19] I. Marson, “Free software’s white knight”, *ZDNet*, 20-mar-2006. [En línea]. Disponible en: http://www.zdnet.com/article/free-softwares-white-knight-5000147301/. [Consultado: 22-ene-2018].

[20] E. Moglen, “Snowden and the Future - Part II: Oh, Freedom”, 30-oct-2013. [En línea]. Disponible en: http://www.snowdenandthefuture.info/PartII.html. [Consultado: 26-ene-2018].
[21] K. Rodríguez, M. Hernández Anzora, H. Sierra-Castro, J. Jiménez Barillas, Tábora Gonzales, Edy, y Zepeda Rivera, Mireya, *¿Privacidad digital para defensores y defensoras de derechos humanos?*, Primera. San José, Costa Rica, 2015.
[22] “La Declaración Universal de Derechos Humanos”. [En línea]. Disponible en: https://www.un.org/es/universal-declaration-human-rights/. [Consultado: 24-jul-2017].
[23] “Constitución del Ecuador”. [En línea]. Disponible en: http://www.asambleanacional.gob.ec/documentos/constitucion_de_bolsillo.pdf.
[24] Fratti Sara, “Informe Anual 2017 - Observatorio Centroamericano de Seguridad Digital”. .
[25] R. Stallman, “¿A quién sirve realmente ese servidor?” [En línea]. Disponible en: https://www.gnu.org/philosophy/who-does-that-server-really-serve.html. [Consultado: 21-feb-2018].
[26] “BSD Overview”. [En línea]. Disponible en: https://developer.apple.com/library/archive/documentation/Darwin/Conceptual/KernelProgramming/BSD/BSD.html. [Consultado: 05-sep-2018].
[27] “The Android Source Code”, *Android Open Source Project*. [En línea]. Disponible en: https://source.android.com/setup/. [Consultado: 05-sep-2018].
[28] D. Greene, “As G Suite gains traction in the enterprise, G Suite’s Gmail and consumer Gmail to more closely align”, *Google*, 23-jun-2017. [En línea]. Disponible en: https://www.blog.google/products/gmail/g-suite-gains-traction-in-the-enterprise-g-suites-gmail-and-consumer-gmail-to-more-closely-align/. [Consultado: 05-sep-2018].
[29] Bergen, Mark y Surane, Jennifer, “AP Exclusive: Google tracks your movements, like it or not”, *AP News*. [En línea]. Disponible en: https://apnews.com/828aefab64d4411bac257a07c1af0ecb. [Consultado: 05-sep-2018].
[30] “Google and Mastercard Cut a Secret Ad Deal to Track Retail Sales”, *Bloomberg.com*, 30-ago-2018.
[31] Stallman, Richard, “El software libre es ahora aún más importante”. [En línea]. Disponible en: https://www.gnu.org/philosophy/free-software-even-more-important.es.html. [Consultado: 05-sep-2018].
[32] “¿A quién sirve realmente ese servidor?” [En línea]. Disponible en: https://www.gnu.org/philosophy/free-software-even-more-important.es.html. [Consultado: 07-feb-2018].
[33] A. Ankerholz, “2017 Linux Kernel Report Highlights Developers’ Roles and Accelerating Pace of Change”, *The Linux Foundation*, 25-oct-2017. .
[34] “Edward Snowden: NSA whistleblower answers reader questions”, *The Guardian*, 17-jun-2013.
[35] “Intercept with PGP encrypted message”. [En línea]. Disponible en: https://edwardsnowden.com/2015/01/06/intercept-with-pgp-encrypted-message/. [Consultado: 22-sep-2017].
[36] “Intercept with OTR encrypted chat”. [En línea]. Disponible en: https://edwardsnowden.com/2015/01/07/intercept-with-otr-encrypted-chat/. [Consultado: 22-sep-2017].
[37] “Different Types of Encryption”. [En línea]. Disponible en: https://sec.eff.org/articles/different-encryption. [Consultado: 22-nov-2018].
[38] P. Higgins, “Pushing for Perfect Forward Secrecy, an Important Web Privacy Protection”, *Electronic Frontier Foundation*, 28-ago-2013. [En línea]. Disponible en: https://www.eff.org/deeplinks/2013/08/pushing-perfect-forward-secrecy-important-web-privacy-protection. [Consultado: 16-oct-2018].

[39] S. Levy, “Crypto Rebels”, *WIRED*, 01-feb-1993. [En línea]. Disponible en: https://www.wired.com/1993/02/crypto-rebels/. [Consultado: 29-ene-2018].
[40] P. Zimmermann, “Phil Zimmermann on PGP”, nov-1994. [En línea]. Disponible en: https://philzimmermann.com/EN/essays/index.html. [Consultado: 29-ene-2018].
[41] T. Rid, “The cypherpunk revolution”, *Atavist*, 20-jul-2016. [En línea]. Disponible en: http://projects.csmonitor.com/cypherpunk. [Consultado: 29-ene-2018].
[42] “The Electronic Frontier Foundation”, 07-may-2017. [En línea]. Disponible en: https://web.archive.org/web/20170507231657/https://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19980716_eff_des_faq.html. [Consultado: 20-nov-2018].
[43] Appelbaum, Jacob *et al.*, “Prying Eyes: Inside the NSA’s War on Internet Security”, *SPIEGEL ONLINE*, 28-dic-2014. [En línea]. Disponible en: http://www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html. [Consultado: 03-ene-2017].
[44] “Exclusive: Secret contract tied NSA and security industry pioneer”, *Reuters*, 20-dic-2013.
[45] https://www.facebook.com/hokietiger88, “NSA seeks to build quantum computer that could crack most types of encryption”, *Washington Post*. [En línea]. Disponible en: https://www.washingtonpost.com/world/national-security/nsa-seeks-to-build-quantum-computer-that-could-crack-most-types-of-encryption/2014/01/02/8fff297e-7195-11e3-8def-a33011492df2_story.html. [Consultado: 20-nov-2018].
[46] J. Assange, “Flujos de información y poder”, *América Latina en movimiento*, 10-abr-2014. [En línea]. Disponible en: http://www.alainet.org/es/articulo/84739. [Consultado: 26-ene-2017].
[47] EFF, “Escogiendo el VPN Apropiado Para Tí”, *Autoprotección Digital Contra La Vigilancia*, 17-sep-2014. [En línea]. Disponible en: https://ssd.eff.org/es/module/escogiendo-el-vpn-apropiado-para-t%C3%AD. [Consultado: 09-mar-2018].
[48] The Tor Project, “Tor Project: Overview”. [En línea]. Disponible en: https://www.torproject.org/about/overview. [Consultado: 07-dic-2017].
[49] R. Dingledine, “[tor-talk] Questions about Directory Authority Servers”, 15-oct-2018.
[50] The Tor Project, “Relay Search”. [En línea]. Disponible en: https://metrics.torproject.org/rs.html#search/flag:authority. [Consultado: 28-feb-2018].
[51] The Tor Project, “Servers – Tor Metrics”. [En línea]. Disponible en: https://metrics.torproject.org/. [Consultado: 04-mar-2018].
[52] M. Leech <mleech@bnr.ca>, “SOCKS Protocol Version 5”. [En línea]. Disponible en: https://tools.ietf.org/html/rfc1928. [Consultado: 04-mar-2018].
[53] R. Dingledine, N. Mathewson, y P. Syverson, “Tor: The second-generation onion router”, DTIC Document, 2004.
[54] K. Finley, “Out in the Open: Inside the Operating System Edward Snowden Used to Evade the NSA”, *WIRED*, 14-abr-2014. [En línea]. Disponible en: https://www.wired.com/2014/04/tails/. [Consultado: 20-nov-2016].
[55] R. Bonifaz, “Acceder a un Computador a través de Tor sin IP Público | Rafael Bonifaz”, 24-feb-2015. [En línea]. Disponible en: https://rafael.bonifaz.ec/blog/2015/02/acceder-a-un-computado-a-traves-de-tor-sin-ip-publico/comment-page-1/#comments. [Consultado: 01-mar-2018].
[56] The Tor Project, “Tor: Onion Service Protocol”. [En línea]. Disponible en: https://www.torproject.org/docs/onion-services. [Consultado: 07-dic-2017].
[57] “A Gentle Introduction to How I2P Works - I2P”. [En línea]. Disponible en: https://geti2p.net/en/docs/how/intro. [Consultado: 01-mar-2018].
[58] “Naming and Addressbook - I2P”. [En línea]. Disponible en: https://geti2p.net/en/docs/naming. [Consultado: 07-mar-2018].
[59] “STARTTLS Everewhere”, *STARTTLS Everewhere*. [En línea]. Disponible en: https://starttls-everywhere.org/about/. [Consultado: 27-ago-2018].

[60] “The GNU Privacy Handbook”. [En línea]. Disponible en: https://www.gnupg.org/gph/en/manual.html. [Consultado: 26-abr-2018].
[61] R. Klafter, “Evil 32: Check Your GPG Fingerprints”. [En línea]. Disponible en: https://evil32.com/. [Consultado: 27-abr-2018].
[62] “Chapter 2. Applying to Become a Maintainer”. [En línea]. Disponible en: https://www.debian.org/doc/manuals/developers-reference/ch02.en.html. [Consultado: 27-ago-2018].
[63] “operational pgp - draft”, *Gist*. [En línea]. Disponible en: https://gist.github.com/grugq/03167bed45e774551155. [Consultado: 01-may-2018].
[64] Protonmail, “ProtonMail is Open Source!”, *ProtonMail Blog*, 13-ago-2015. .
[65] B. Butler, “ProtonMail now the maintainer of OpenPGPjs email encryption library”, *ProtonMail Blog*, 02-ago-2016. .
[66] Andy Yen, “Introducing Address Verification and Full PGP Support”, *ProtonMail Blog*, 25-jul-2018. .
[67] B. Wolford, “What is zero-access encryption and why it is important for security”, *ProtonMail Blog*, 23-may-2018. .
[68] Protonmail, “What is One Password Mode?”, *ProtonMail Support*. .
[69] Protonmail, “The ProtonMail Threat Model”, *ProtonMail Blog*, 19-may-2014. .
[70] D. Stenberg, “History of IRC (Internet Relay Chat)”, 29-mar-2011. [En línea]. Disponible en: https://daniel.haxx.se/irchistory.html. [Consultado: 15-oct-2018].
[71] J. Oikarinen y D. Reed, “Internet Relay Chat Protocol”, may-1993. [En línea]. Disponible en: https://tools.ietf.org/html/rfc1459#section-1. [Consultado: 15-oct-2018].
[72] “XMPP | An Overview of XMPP”. [En línea]. Disponible en: https://xmpp.org/about/technology-overview.html. [Consultado: 19-nov-2016].
[73] Constantine, Josh, “WhatsApp hits 1.5 billion monthly users. $19B? Not so bad.”, *TechCrunch*, 13-ene-2018. .
[74] Durov, Pavel, “200,000,000 Monthly Active Users”, *200,000,000 Monthly Active Users*, 22-mar-2018. [En línea]. Disponible en: https://telegram.org/blog/200-million. [Consultado: 13-sep-2018].
[75] “Registro la línea de mi celular”, *Argentina.gob.ar*, 04-oct-2017. [En línea]. Disponible en: https://www.argentina.gob.ar/justiciacerca/registrolineacelular. [Consultado: 18-sep-2018].
[76] “Los chips no funcionan sin registro de datos reales”, *El Universo*, 06-may-2018. [En línea]. Disponible en: https://www.eluniverso.com/noticias/2018/05/06/nota/6745604/chips-no-funcionan-registro-datos-reales. [Consultado: 18-sep-2018].
[77] P. Higgins, “Google Abandons Open Standards for Instant Messaging”, *Electronic Frontier Foundation*, 22-may-2013. [En línea]. Disponible en: https://www.eff.org/es/deeplinks/2013/05/google-abandons-open-standards-instant-messaging. [Consultado: 15-oct-2018].
[78] “mod_onions - Prosody Community Modules”. [En línea]. Disponible en: http://modules.prosody.im/mod_onions.html. [Consultado: 20-sep-2018].
[79] Alkemade, Thijs, “XMPP federation over Tor hidden services”, 12-abr-2013. [En línea]. Disponible en: https://blog.thijsalkema.de/blog/2013/04/02/xmpp-federation-over-tor-hidden-services/. [Consultado: 20-sep-2018].
[80] “Off-the-Record Messaging Protocol version 3”. [En línea]. Disponible en: https://otr.cypherpunks.ca/Protocol-v3-4.1.1.html. [Consultado: 30-mar-2018].
[81] B. B. and G. Gebhart, “Where WhatsApp Went Wrong: EFF’s Four Biggest Security Concerns”, *Electronic Frontier Foundation*, 13-oct-2016. [En línea]. Disponible en: https://www.eff.org/deeplinks/2016/10/where-whatsapp-went-wrong-effs-four-biggest-security-concerns. [Consultado: 21-sep-2018].
[82] S. Celi, “[OTR-dev] OTR version 4 Draft #2”, 16-mar-2018.

[83] “Acerca”. [En línea]. Disponible en: https://autonomia.digital/es/about/. [Consultado: 12-nov-2018].
[84] “Signal >> Home”. [En línea]. Disponible en: https://www.signal.org/. [Consultado: 02-abr-2018].
[85] Lee, Micah, “How to Use Signal Without Giving Out Your Phone Number”. [En línea]. Disponible en: https://theintercept.com/2017/09/28/signal-tutorial-second-phone-number/. [Consultado: 24-sep-2018].
[86] J. Lund, “Signal >> Blog >> Signal partners with Microsoft to bring end-to-end encryption to Skype”, 11-ene-2018. [En línea]. Disponible en: https://signal.org/blog/skype-partnership/. [Consultado: 01-abr-2018].
[87] M. Marlinspike, “Signal >> Blog >> Forward Secrecy for Asynchronous Messages”, 22-ago-2013. [En línea]. Disponible en: https://signal.org/blog/asynchronous-security/. [Consultado: 01-abr-2018].
[88] M. Marlinspike, “Signal >> Blog >> Private Group Messaging”, 05-may-2014. [En línea]. Disponible en: https://signal.org/blog/private-groups/. [Consultado: 05-abr-2018].
[89] M. Marlinspike, “Signal >> Blog >> Reproducible Signal builds for Android”. [En línea]. Disponible en: https://signal.org/blog/reproducible-android/. [Consultado: 25-sep-2018].
[90] M. Marlinspike, *Contribute to signalapp/Signal-Server development by creating an account on GitHub*. Signal, 2018.
[91] S. Verschoor, “OMEMO: Cryptographic Analysis Report”. 01-jun-2016.
[92] “Ricochet”, *Ricochet*. [En línea]. Disponible en: https://ricochet.im/. [Consultado: 20-nov-2016].
[93] T. Grote, *Briar - Resilient P2P Messaging for Everyone*. Chaos Communication Congress, 2017.
[94] “Briar User Manual”. [En línea]. Disponible en: https://briarproject.org/manual/. [Consultado: 26-sep-2018].
[95] “Por qué Delta Chat? Tus ventajas. - Delta Chat”. [En línea]. Disponible en: https://delta.chat/es/features. [Consultado: 26-sep-2018].
[96] “Autocrypt FAQ — Autocrypt 1.0.1 documentation”. [En línea]. Disponible en: https://autocrypt.org/faq.html. [Consultado: 25-oct-2018].
[97] “Autocrypt Level 1: Enabling encryption, avoiding annoyances — Autocrypt 1.0.1 documentation”. [En línea]. Disponible en: https://autocrypt.org/level1.html. [Consultado: 25-oct-2018].
[98] E. Schooler *et al.*, “SIP: Session Initiation Protocol”. [En línea]. Disponible en: https://tools.ietf.org/html/rfc3261. [Consultado: 26-oct-2018].
[99] V. Jacobson, R. Frederick, S. Casner, y H. Schulzrinne, “RTP: A Transport Protocol for Real-Time Applications”. [En línea]. Disponible en: https://tools.ietf.org/html/rfc3550#section-11. [Consultado: 26-oct-2018].
[100] D. Wing, P. Matthews, R. Mahy, y J. Rosenberg, “Session Traversal Utilities for NAT (STUN)”. [En línea]. Disponible en: https://tools.ietf.org/html/rfc5389. [Consultado: 26-oct-2018].
[101] D. A. McGrew y K. Norrman, “The Secure Real-time Transport Protocol (SRTP)”. [En línea]. Disponible en: https://tools.ietf.org/html/rfc3711#page-30. [Consultado: 26-oct-2018].
[102] P. Zimmermann, J. Callas, y A. Johnston, “ZRTP: Media Path Key Agreement for Unicast Secure RTP”. [En línea]. Disponible en: https://tools.ietf.org/html/rfc6189#page-77. [Consultado: 26-oct-2018].
[103] P. Saint-Andre, “Use of ZRTP in Jingle RTP Sessions”, 15-jun-2011. [En línea]. Disponible en: https://xmpp.org/extensions/xep-0262.html. [Consultado: 29-oct-2018].
[104] “FAQ – OnionCat”. [En línea]. Disponible en: https://www.onioncat.org/faq/. [Consultado: 29-oct-2018].

[105] “User guide - Linphone open source video sip phone, voip software”. [En línea]. Disponible en: https://www.linphone.org/user-guide.html. [Consultado: 30-oct-2018].
[106] “Technical aspects of Ring”, *Ring*, 09-nov-2015. [En línea]. Disponible en: https://ring.cx/en/about/technical. [Consultado: 30-oct-2018].
[107] “About Ring”, *Ring*, 06-nov-2015. [En línea]. Disponible en: https://ring.cx/en/about/practical. [Consultado: 30-oct-2018].
[108] G. Roguez, “remove dead code about ZRTP (17ca5194) · Commits · savoirfairelinux / ring-lrc”, *GitLab*, 19-ago-2016. [En línea]. Disponible en: https://git.ring.cx/savoirfairelinux/ring-lrc/commit/17ca519496c7b4bafd57ef5495fe57638d9a7631. [Consultado: 30-oct-2018].
[109] “Privacy and anonymity”, *Privacy and Anonymiti* | *Ring*, 23-ago-2016. [En línea]. Disponible en: https://ring.cx/en/about/privacy-and-anonymity. [Consultado: 16-may-2018].
[110] “Tox Clients”, *Project Tox*. [En línea]. Disponible en: https://tox.chat/clients.html. [Consultado: 27-sep-2018].
[111] “FAQ - Tox”, *Project Tox*. [En línea]. Disponible en: https://tox.chat/faq.html. [Consultado: 09-may-2018].
[112] “FAQ/English - Mumble Wiki”. [En línea]. Disponible en: https://wiki.mumble.info/wiki/FAQ. [Consultado: 16-may-2018].

Índice de Imágenes

Imágenes

Imagen 1: Diapositiva de presentación sobre PRISM de 2013....4
Imagen 2: Formas en que la NSA recolecta información....5
Imagen 3: Hacking Team en América Latina....8
Imagen 4: Modelo de amenaza 11
Imagen 5: Red Privada Virtual - VPN....25
Imagen 6: Ruteo cebolla....26
Imagen 7: Navegador Tor....28
Imagen 8: Publicación del servicio oculto....31
Imagen 9: Servicios ocultos.- punto de encuentro 32
Imagen 10: Servicios Ocultos.- canal establecido....33
Imagen 11: I2P: Túneles....34
Imagen 12: Panel de control de I2P....35
Imagen 13: Correo secreto con seudónimos....42
Imagen 14: Protecciones contra abuso de Protonmail.....45
Imagen 15: Mensaje cifrado con clave simétrica con Protonmail....45
Imagen 16: Correo federado con servicios cebolla....48
Imagen 17: Acceso a *webmail* mediante servicio oculto de Tor....49
Imagen 18: Federación en XMPP....54
Imagen 19: Crear cuenta de chat desde CoyIM....56
Imagen 20: Interfaz administrativa de Ejabberd....57
Imagen 21: Configuración Conversations y servicio cebolla....58
Imagen 22: Verificación de mensajes de OTR.....60
Imagen 23: Autenticación OTR....61
Imagen 24: Mensaje enviado a un chat grupal de Signal.....62
Imagen 25: Funcionamiento OMEMO....65
Imagen 26: Ricochet cliente P2P Fuente....67
Imagen 27: Captura de pantalla Ricochet.....68
Imagen 28: Presentación de contactos en Briar....69
Imagen 29: Verificación de llaves con Delta Chat....71
Imagen 30: Ping6 con Onioncat.....76
Imagen 31: Esquema de Mumble sin Tor....79
Imagen 32: Mumble como servicio cebolla....80
Imagen 33: Configuración clientes Mumble....81